

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle
En vue de l'obtention du Diplôme de Master Professionnel en Informatique
Option:
Génie Logiciel

Thème:

Conception et Implémentation d'un MVP d'une plateforme de Data Visualisation

Présenté par:

Mr KEBBI Billal

Mr MEHIDI Anis

Évalué le 14 octobre 2021 par le jury composé de

Président de jury :	KHENOUCHE Med Essaid	M.C.A
Examineur :	ALOUÏ Soraya	M.C.A
Encadrant :	KHOULALENE Nadjat	M.C.B

Année universitaire : 2020/2021

Dédicaces

Ce modeste travail est dédié : A nos chers parents, pour tous leurs sacrifices, leurs amour, leurs tendresse, leurs soutien et leurs prières tout au long de nos études.

A nos chers frères et soeurs, pour leur appui et leurs encouragements permanents, et leur soutien moral tout au long de notre parcours universitaire.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible.

Merci d'être toujours là pour nous.

Remerciements

*Au terme de ce projet de fin d'études, nous tenons à remercier dans un premier temps le Gérant - fondateur de **Tech Instinct Mr. BELATTAF Youcef** et tout le personnel pour leur accueil, leur confiance et de nous avoir fait découvrir et approché du monde de l'informatique.*

*Nous voudrions également remercier notre encadrant **Mme. KHOULALENE**, de nous avoir encadré.*

*Nous tenons à exprimer toute notre reconnaissance à notre co-encadrant de mémoire **Mr Kherib Chafaa**, MOA (chef de projet) à Tech Instinct, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter nos réflexions.*

*Nos remerciements s'adressent également à **Mr. MEHDI Braham**, développeur Front End à Tech Instinct, qui a su se rendre disponibles pour répondre à nos questions, suivre notre travail et aussi pour son encouragement, ouverture d'esprit, et surtout sa bonne humeur.*

Nos remerciements vont pareillement aux membres du jury pour avoir accepté d'examiner notre travail et de l'enrichir par leurs propositions.

*Nous remercions toute l'équipe pédagogique du département d'informatique de l'université **ABDERRAHMANE MIRA**.*

Ainsi qu'à toutes les personnes qui ont contribué au succès de notre stage et qui nous ont aidés lors de la rédaction de ce mémoire.

Table des matières

Table des matières	iv
Table des figures	v
Liste des tableaux	1
Glossaire	2
Introduction générale	3
1 État de l’art sur la Data visualisation	6
1.1 Introduction	6
1.2 Présentation de la visualisation de données	7
1.2.1 Définitions	7
1.2.2 Les caractéristiques d’une bonne visualisation de données	8
1.3 État de l’art sur les outils de la visualisation des données	9
1.3.1 Visualisation prête à l’emploi	9
1.3.2 Tableau comparatif entre les outils de data visualisation	16
1.3.3 Programmation d’un logiciel de visualisation	16
1.4 Enjeux et opportunités de la visualisation des données	17
1.5 Tableaux de bord (Dashboard)	17
1.6 Mise en forme des Tableaux de bord	18
1.6.1 Les différents types de graphiques	18
1.6.2 La notion du design	21
1.6.3 Data visualisation et infographie	22
1.7 Conclusion	23
2 Analyse & spécification des besoins	24

2.1	Introduction	24
2.2	Présentation de l'organisme d'accueil	24
2.2.1	Présentation de Tech Instinct	24
2.2.2	Domaines d'expertise et technologies adoptées	25
2.3	Problématique et motivation	25
2.4	Besoins de l'entreprise d'accueil	26
2.4.1	Besoins fonctionnels	26
2.4.2	Besoins non fonctionnels	27
2.5	Identification des acteurs	27
2.6	Identification des cas d'utilisation	28
2.6.1	Diagramme de cas d'utilisation	30
2.6.2	Description textuelle des cas d'utilisation	32
2.7	Conclusion	44
3	Conception	45
3.1	Introduction	45
3.2	Diagramme de séquence système	45
3.2.1	Diagramme de séquence Authentification	45
3.2.2	Diagramme de séquence Gestion des projets	46
3.2.3	Diagramme de séquence Gestion des Dashboard	48
3.2.4	Diagramme de séquence Gestion des Graphes	49
3.2.5	Diagramme de séquence Gestion des Comptes	50
3.2.6	Diagramme de séquence Gestion visiteurs	51
3.3	Diagramme de classes	52
3.4	Modèle relationnel	53
3.5	Dictionnaire de données	56
3.6	Conclusion	57
4	Réalisation	58
4.1	Introduction	58
4.2	Application web	58
4.3	Outils de développement	59
4.3.1	Visual studio code	59
4.3.2	Visual Paradigm	59
4.3.3	Wampserver	59
4.4	Langages de programmation	59

4.4.1	Bootstrap	60
4.4.2	MySQL (My Structured Query Language)	60
4.4.3	Git	60
4.4.4	JavaScript	60
4.4.5	HTML (Hypertext Markup Language)	60
4.4.6	CSS	61
4.4.7	Angular	61
4.4.8	TypeScript	61
4.4.9	Ngx-Charts	61
4.4.10	Swimlane	62
4.4.11	NodeJs	62
4.4.12	Node Package Manager (NPM)	62
4.4.13	JSON	62
4.4.14	Express	63
4.5	Notre application de gestion de la DataViz	63
4.5.1	Interface d'authentification	63
4.5.2	Interface Inscription	64
4.5.3	Gestion des Utilisateurs	65
4.5.3.1	Interface " liste des utilisateurs "	65
4.5.3.2	Interface " Ajouter un utilisateur "	65
4.5.3.3	Interface " Modifier un utilisateur "	66
4.5.3.4	Interface " Supprimer un utilisateur "	67
4.5.4	Gestion des Projets	67
4.5.4.1	Interface " liste des projets "	67
4.5.4.2	Interface " Ajouter un projet "	68
4.5.4.3	Interface " Modifier un projet "	68
4.5.4.4	Interface " Supprimer un projet "	69
4.5.5	Gestion des Dashboards	70
4.5.5.1	Interface " liste des Dashboards "	70
4.5.5.2	Interface " Ajouter un Dashboard "	70
4.5.5.3	Interface " Modifier un Dashboard "	71
4.5.5.4	Interface " Supprimer un Dashboard "	71
4.5.6	Gestion des Graphes	72
4.5.6.1	Interface " Ajouter un graphe dans un Dashboard "	72
4.5.6.2	Interface " Visualisation des graphes dans un Dashboard "	72
4.6	Conclusion	73

TABLE DES MATIÈRES

Conclusion générale	74
Bibliographie	76

Table des figures

1.1	Feuille de calcul Excel.	10
1.2	Options des graphiques Excel	11
1.3	Google Sheet	12
1.4	Tableau de bord de Chartio.	13
1.5	Tableau de bord de Holistics.	14
1.6	Exemple de tableau de bord de Tableau.	15
1.7	Exemple de tableau de bord Klipfolio.	16
1.8	Histogramme	18
1.9	Graphique linéaire	19
1.10	Graphique à double accès	19
1.11	Graphique à secteurs	20
1.12	Graphique Nuage de points	20
1.13	Graphique à bulles	21
2.1	Logo entreprise « Tech Instinct ».	25
2.2	Diagramme de cas d'utilisation globale du système.	31
2.3	Diagramme de cas d'utilisation « Authentification ».	33
2.4	Diagramme de cas d'utilisation « Consultation de la vidéo démonstrative ».	34
2.5	Diagramme de cas d'utilisation « Gestion des projets ».	37
2.6	Diagramme de cas d'utilisation « Gestion des Dashboard ».	39
2.7	Diagramme de cas d'utilisation « Gestion des graphes dans les Dashboard ».	42
2.8	Diagramme de cas d'utilisation « Gestion des comptes ».	44
3.1	Diagramme de séquence Authentification.	46
3.2	Diagramme de séquence Gestion des projets.	47
3.3	Diagramme de séquence Gestion des Dashboard.	48
3.4	Diagramme de séquence Gestion des graphes.	49
3.5	Diagramme de séquence Gestion des comptes.	50
3.6	Diagramme de séquence Gestion visiteurs.	51

3.7	Diagramme de classes.	52
3.8	Relation un à plusieurs.	53
3.9	Relation plusieurs à plusieurs.	54
3.10	Relation un à un.	54
3.11	Relation généralisation.	55
4.1	Interface d'authentification.	64
4.2	Interface d'inscription.	64
4.3	Interface " liste des utilisateurs ".	65
4.4	Interface " Ajouter un utilisateur ".	66
4.5	Interface " Modifier un utilisateur ".	66
4.6	Interface " Supprimer un utilisateur ".	67
4.7	Interface " liste des projets ".	67
4.8	Interface " Ajouter un projet ".	68
4.9	Interface " Modifier un projet ".	69
4.10	Interface " Supprimer un projet ".	69
4.11	Interface " liste des Dashboards ".	70
4.12	Interface " Ajouter un Dashboard ".	70
4.13	Interface " Modifier un Dashboard ".	71
4.14	Interface " Supprimer un Dashboard ".	71
4.15	Interface " Ajouter un graphe dans un Dashboard ".	72
4.16	Interface " Visualisation des graphes dans un Dashboard ".	73

Liste des tableaux

1.1	Comparaison entre les outils de data visualisation.	16
2.1	Identification des cas d'utilisation.	29
2.2	Description du cas d'utilisation « Authentification ».	32
2.3	Description du cas d'utilisation « Consultation de la vidéo démonstrative ».	34
2.4	Description du cas d'utilisation « Gestion des projets ».	36
2.5	Description du cas d'utilisation « Gestion des Dashboard ».	39
2.6	Description du cas d'utilisation « Gestion des graphes dans les Dashboard ».	41
2.7	Description du cas d'utilisation « Gestion des comptes ».	43
3.1	Dictionnaire de données.	57

Glossaire

Amazon RedShift est un produit d'entrepôt de données qui fait partie de la plus grande plate- forme de cloud computing Amazon Web Services. **page.** 12

Angular Développé par Google, Angular est un Framework open source écrit en JavaScript qui permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. **page.** 16

Business intelligence (en français L'informatique décisionnelle) est l'informatique à l'usage des décideurs et des dirigeants d'entreprises. Elle désigne les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données, matérielles ou immatérielles, d'une entreprise en vue d'offrir une aide à la décision et de permettre à un décideur d'avoir une vue d'ensemble de l'activité traitée. **page.** 12

Google Analytics est un service gratuit d'analyse d'audience d'un site Web ou d'applications utilisé par plus de 10 millions de sites, soit plus de 80% du marché mondial. **page.** 12

KPI (l'indicateur de performance clé) est une valeur mesurable qui démontre l'efficacité avec laquelle une entreprise atteint ses objectifs commerciaux clés. **page.** 17

Microsoft Azure est la plate-forme applicative en nuage de Microsoft. Son nom évoque le « cloud computing », ou informatique en nuage. **page.** 12

MySQL est un système de gestion de bases de données relationnelles (SGBDR). **page.** 12

PHP Hypertext Preprocessor, plus connu sous son sigle PHP est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. **page.** 16

Phyton est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. **page.** 16

Vert.x est un framework événementiel pour la JVM s'inspirant librement de Node.js. Il est publié en open source sous licences Apache Software Licence 2.0 et Eclipse Public Licence. En particulier, Vert.x permet la création de serveurs web, permettant notamment une montée en charge très efficace. **page.** 16

Introduction générale

La technologie est aujourd'hui un élément fondamental pour toute entreprise. L'évolution dans le domaine de l'informatique offre des opportunités pour améliorer les processus de flux des travaux. Elle permet de faire circuler l'information dans toute l'entreprise grâce aux logiciels et devient donc un outil de gestion et de pilotage de l'entreprise. De nos jours, les bases de données des entreprises sont de plus en plus volumineuses, il s'agit d'une explosion quantitative de la donnée numérique. Cette richesse de données peut-être une mine d'informations pour aider à l'amélioration de la prise de décision, à la mise en évidence de constats, à la transmission d'idées plus claires, et à avoir un regard objectif. C'est pourquoi la plupart des entreprises se tournent vers la Data Visualisation, comme c'est le cas de Tech'Instinct, l'entreprise où nous avons effectué notre stage.

La visualisation des données est une représentation graphique de toute donnée ou information. Les éléments visuels tels que les tableaux, les graphiques et les cartes sont les quelques outils de visualisation de données qui nous offre un moyen simple et accessible de comprendre les informations représentées.

Il existe une multitude de plateforme de visualisation des données, chacune a ses avantages mais aussi ses inconvénients, car la plupart de ces outils coûtent cher et peuvent ne pas satisfaire tout les besoins du client

C'est pourquoi nous avons décidé de créer une plateforme de Dataviz qui permet de répondre aux problématiques de l'entreprise tech-instinct vis-à-vis l'exploitation et la visualisation des données, voire apporter de nouveaux usages, d'avoir son propre outil de visualisation de données au lieu de payer des abonnements pour ce service et également de commercialiser cette plateforme car cette dernière est une entreprise qui développe des solutions informatique.

Cette application offre à l'entreprise un moyen pour mieux visualiser ses données à travers des tableaux de bord (dashboards) avec de multiples graphiques mais aussi la gestion des projets et aussi la gestion des utilisateurs de l'application.

Notre mémoire est structuré en 4 chapitres :

- Le premier chapitre est consacré à la présentation des différentes définitions de la data visualisation selon plusieurs experts, les caractéristiques d'une bonne data visualisation, un état de l'art sur les outils de la visualisation des données déjà existants, les enjeux et opportunités de la visualisation des données, et la définition d'un tableau de bord de graphiques et la mise en forme du tableau de bord.
- Dans le second chapitre, nous présenterons l'entreprise d'accueil Tech'Instinct, suivie de la problématique ainsi que les besoins de l'entreprise d'accueil, ensuite nous allons résumer les acteurs qui interagissent avec le système à développer, ainsi que les cas d'utilisation correspondant à chaque acteur.
- Le troisième chapitre sera consacré à la conception du système en élaborant les diagrammes de séquence système ainsi que le diagramme de classe, le modèle relationnel et le dictionnaire de données.
- Le quatrième et dernier chapitre est consacré à la réalisation de notre application que nous avons réalisé. Nous présenterons les différents outils et langages utilisés ainsi que l'architecture des Frameworks et les technologies utilisées et quelques interfaces de l'application.

Enfin, le mémoire s'achève par une conclusion générale qui résume l'essentiel de notre travail ainsi que quelques perspectives.

État de l'art sur la Data visualisation

1.1 Introduction

Quand on parle de "données", on s'aperçoit que c'est particulièrement ambigu. Par exemple dans une entreprise on s'interroge sur le rôle des données, des chiffres, des indicateurs, des tableaux . . . Il est vrai que les données sont aujourd'hui omniprésentes et apparaissent comme une évidence. Pourtant, elles sont très difficiles à appréhender. Les données qui s'offrent ainsi aux professionnels pour les guider dans leurs décisions sont de plus en plus nombreuses et variées. Mais comment ne pas être submergé et en faire un réel outil pour la réflexion et la prise de décision ? C'est face à cette nécessité que la data visualisation apparaît comme une solution. La représentation des données sous forme d'images pourrait permettre de mieux les comprendre. Le monde des chiffres et celui des images sont a priori très éloignés. Mais comme souvent, c'est en abattant des a priori que l'on progresse.

Ce chapitre a pour but, de faire découvrir le monde de la Data visualisation, d'en poser les fondamentaux, de donner un aperçu des possibilités, des enjeux, et des outils de la visualisation des données . . . Pour ce faire, nous nous appuyerons sur des exemples concrets et des études de cas d'acteurs reconnus dans le domaine.

1.2 Présentation de la visualisation de données

1.2.1 Définitions

Qu'on l'écrive à l'américaine *data visualization*, en français "datavisualisation" ou familièrement sous son petit nom de "dataviz", la représentation graphique de données statistiques est incontestablement à la mode. Et comme tous les mots à la mode, elle devient un terme fourre-tout, dont on finit par ne plus très bien savoir ce qu'il signifie.

Il n'y a pas encore de définition officielle de la *data visualization*, et chaque spécialiste y va de sa propre description. En voici certaines définitions :

- Statistical Analysis System (SAS), un leader mondial du décisionnel et des solutions de business analytics, définit la *data visualization* comme l'exploration visuelle et interactive et la représentation graphique des données, quelles qu'en soient la volumétrie (des *small data* aux *big data*), la nature ou la provenance. Elle permet aux utilisateurs métier de détecter des phénomènes ou des tendances invisibles de prime abord [1].
- Federica Minichiello Responsable du Laboratoire numérique de l'éducation (LNE) et directrice par intérim du Centre de ressources et d'ingénierie documentaires Greater Paris , définit la *data visualization* comme une partie intégrante d'un processus d'analyse des données et offre à l'utilisateur la possibilité d'explorer des chiffres, en lui proposant des clés pour une lecture critique et, parfois, l'opportunité de découvrir des éléments inattendus [2].
- Guillaume Deschamps responsable d'analyse de données chez STMicroelectronics définit la *data visualization* comme un système permettant, via le support de représentations graphiques, une prise de décision factuelle et plus rapide [1].

En résumé, la *data visualization* sert à présenter rapidement les données et à les rendre lisibles. Elle permet d'analyser et de faciliter la compréhension des données en les retranscrivant efficacement sous une forme visuelle. Une *data visualization* réussie rend simple et assimilable un sujet complexe. Elle permet de créer une représentation rendant les valeurs lisibles graphiquement. Et ce dans le but de faire « parler les données ». La *data visualization* va rendre possible et faciliter l'exploration dans la profondeur des données. Elle ne se limite pas aux chiffres, elle peut être utilisée pour analyser des textes. L'interactivité a porté la *data visualization* dans une nouvelle dimension. Le lecteur, qui par le passé était un simple spectateur passif, est devenu aujourd'hui grâce à la *dataviz* interactive, un lecteur actif qui explore l'information comme il l'entend, en faisant ses propres choix de consultation, dans un ordre qu'il définit lui-même selon ses centres d'intérêt.

1.2.2 Les caractéristiques d'une bonne visualisation de données

C'est l'accessibilité et la lisibilité des informations qui font que la dataviz est réussie. Plus c'est simple, plus c'est efficace. Mais cela ne veut pas dire que c'est facile à faire. Il est possible de ne pas être familiarisé avec certains types de représentation et que le graphique semble déroutant. Mais si la dataviz est correctement titrée et légendée (certaines dataviz complexes proposent un tutoriel pour guider le lecteur), nous devrions pouvoir l'explorer, et découvrir l'histoire qu'elle dévoile. Le graphisme de la visualisation est là pour servir les données et pour en faciliter la compréhension, si c'est esthétique, tant mieux. Car cela peut aider à capter le regard et l'intérêt du lecteur. Les graphistes inventent constamment de nouvelles mises en forme des données. Mais attention aux démonstrations graphiques qui desservent la narration en voulant en faire trop : l'équilibre entre le fond et la forme doit être respecté pour préserver une bonne compréhension[3].

Il existe deux types de data visualisation :

A. La data visualisation classique

Les représentations de diagrammes classiques, comme les graphiques sectoriels (cammemberts), les barres et les courbes ont tellement été utilisées de manière systématique et répétitive qu'elles n'accrochent plus l'œil du lecteur. Une impression de déjà-vu s'est peu à peu installée [3].

B. La data visualisation moderne

Il existe heureusement beaucoup d'autres possibilités pour mettre en forme les données. Elles permettent de faire ressortir l'information et de rendre les graphiques plus attractifs, plus engageants et, selon l'objectif, plus parlants : Les treemaps, les sankey diagrammes, les colonnes en triangles, les bubbles, les graphiques en arc de cercle ou radiaux, les barres ou les courbes circulaires, sans oublier les pictogrammes, etc. Chacune de ces représentations peut être utilisée, il suffit de savoir exactement ce que l'on veut montrer : une comparaison des valeurs par catégories, des évolutions dans le temps, la distribution des données, la visualisation d'une valeur dans un ensemble, l'illustration des relations ou des corrélations entre des variables...

Il y a toujours une représentation graphique plus pertinente que les autres pour faciliter la lecture et la compréhension de l'information, il s'agit de trouver le bon équilibre entre le volume du texte et la meilleure forme visuelle pour retranscrire au mieux l'information [3].

1.3 État de l'art sur les outils de la visualisation des données

Les utilisateurs de la data visualisation sont amenés à faire un choix important quand ils doivent choisir le bon logiciel pour visualiser leurs données, car ces logiciels sont nombreux. Certains logiciels, de type glisser-déplacer, sont prêts à l'emploi. D'autres nécessitent un peu de programmation. Cependant, il existe aussi des outils qui n'ont pas été conçus spécifiquement pour les graphiques de données, mais qui se révèlent néanmoins utiles.

Le présent chapitre traite ces différentes options. Mieux nous savons utiliser les outils de visualisation et en tirer parti, moins nous risquons de nous retrouver à ne savoir que faire d'un ensemble de données et plus nous avons de chances de créer un graphique correspondant [3].

1.3.1 Visualisation prête à l'emploi

Les solutions prêtes à l'emploi sont de loin les plus aisées pour les novices. Il suffit d'utiliser les techniques COPIER ET COLLER pour transférer quelques données ou de charger un fichier CSV, puis sélectionner le graphique de notre choix et modifier éventuellement quelques options pour adapter le résultat selon les besoins .

Ces outils, prêts à l'emploi, varient selon l'application pour laquelle ils ont été conçus. Certaines applications, comme Excel, ou encore Google Sheets, sont destinées à la gestion des données et aux graphiques de base, tandis que d'autres prennent en charge des analyses plus approfondies ainsi que l'exploration visuelle [3].

A. Microsoft Office Excel

La figure 1.1 illustre la feuille de calcul d'Excel, dans laquelle nous allons insérer les données.

Il suffit de cliquer sur le bouton représentant un graphique en barres afin de créer le graphique de notre choix (figure 1.2) : graphique en barres, linéaire, en camembert ou nuage de points [3].

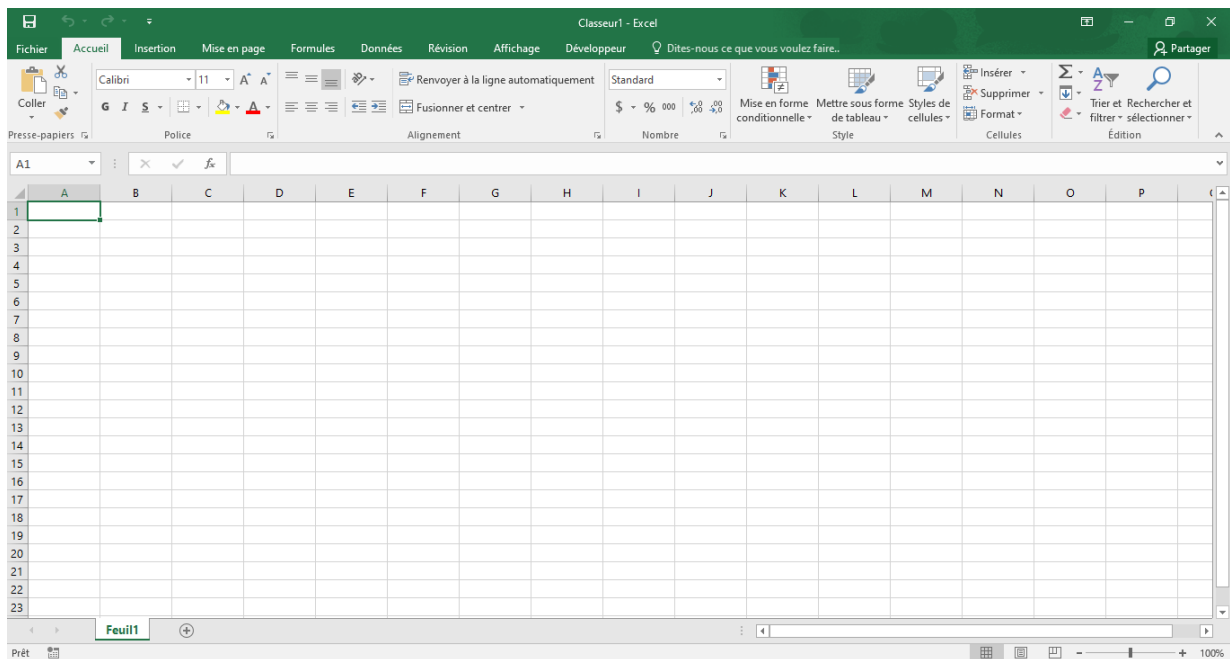


FIGURE 1.1 – Feuille de calcul Excel.

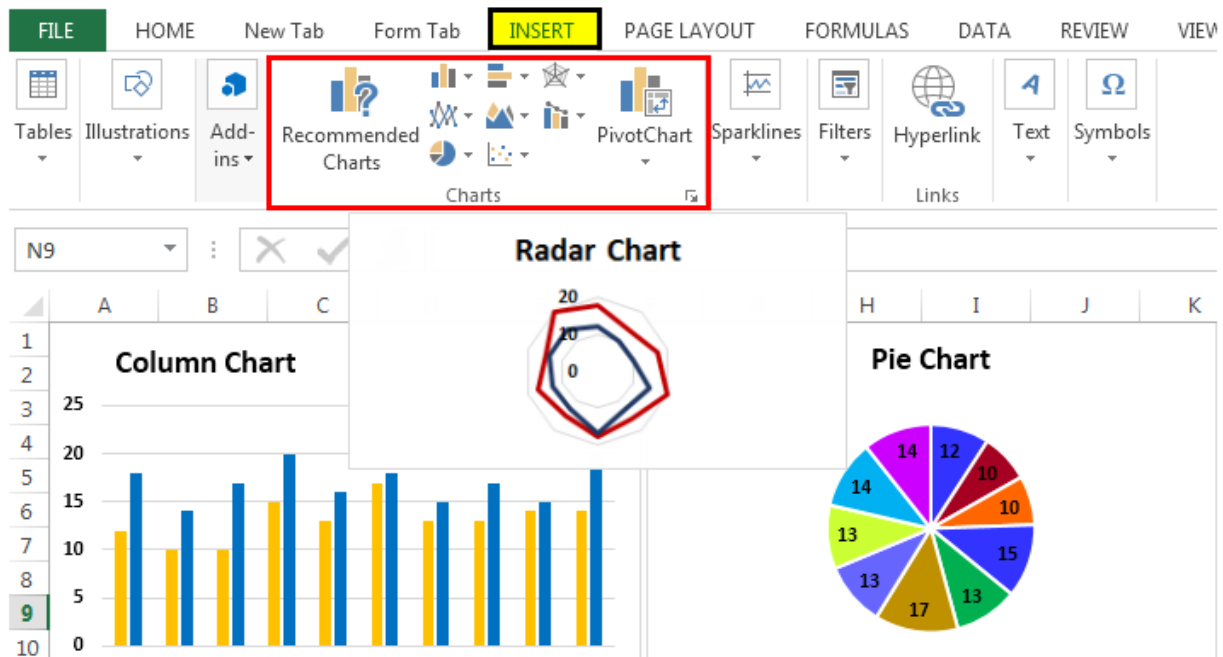


FIGURE 1.2 – Options des graphiques Excel

Certaines personnes dédaignent Excel, alors que c'est une application assez efficace pour les tâches simples. Néanmoins, Excel reste limité en termes de sources de données, qui sont des tableaux que l'utilisateur saisi manuellement dans les feuilles de calculs, et aussi il ne permet pas de faire les analyses approfondies ou des graphiques destinés à une publication.

B. Google sheets

Google Sheets correspond plus ou moins à la version cloud d'Excel, et reprend l'interface familière de ce dernier (figure 1.3). L'application propose également les types de graphiques standards et offre toutefois certains avantages par rapport à Excel. Tout d'abord, comme les données sont stockées sur les serveurs Google, il est possible de les visualiser sur n'importe quel ordinateur, sous réserve qu'un navigateur web y soit installé. Il est également possible de partager aisément une feuille de calcul et collaborer en temps réel [3].

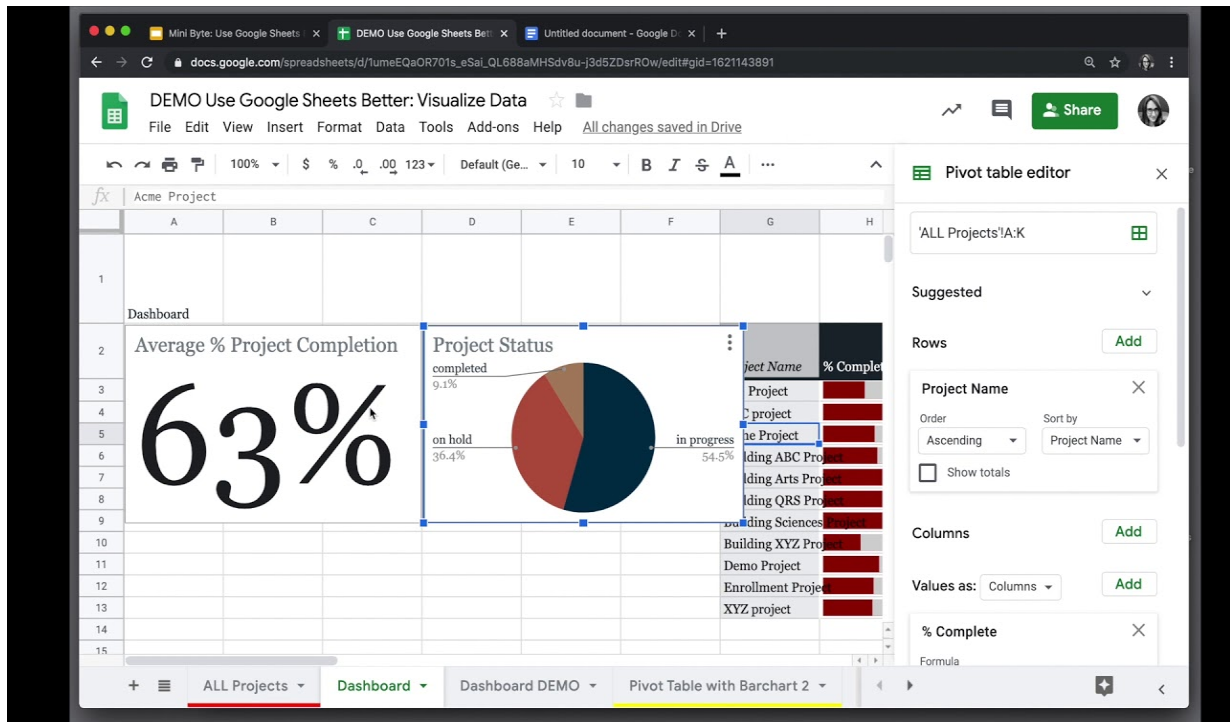


FIGURE 1.3 – Google Sheet

C. Chartio

Chartio a été conçu pour permettre aux startups et aux marketers de mieux exploiter les données. Cet outil peut être relié à d'autres services notamment Microsoft Azure, MySQL, Amazon RedShift, Google Analytics et bien d'autres encore. Il n'y a aucun souci pour la prise en main grâce aux vidéos en ligne, cela permet de gagner du temps dans la création des rapports et se concentrer sur l'analyse des données qui compte pour l'entreprise. Chartio est excellent pour la Business intelligence en libre-service avec les petites et moyennes entreprises. Il propose de multitudes graphes où il est possible de fusionner, superposer et transformer plusieurs ensembles de données en une série d'étapes, il est aussi possible de partager des tableaux de bord via des e-mails ou des rapports planifiés et afficher les données historiques et suivre les mesures de l'entreprise [4].

Le seul inconvénient c'est son prix, car il est à hauteur de 150 dollars par mois par utilisateur ou 249 dollars par mois par équipe.

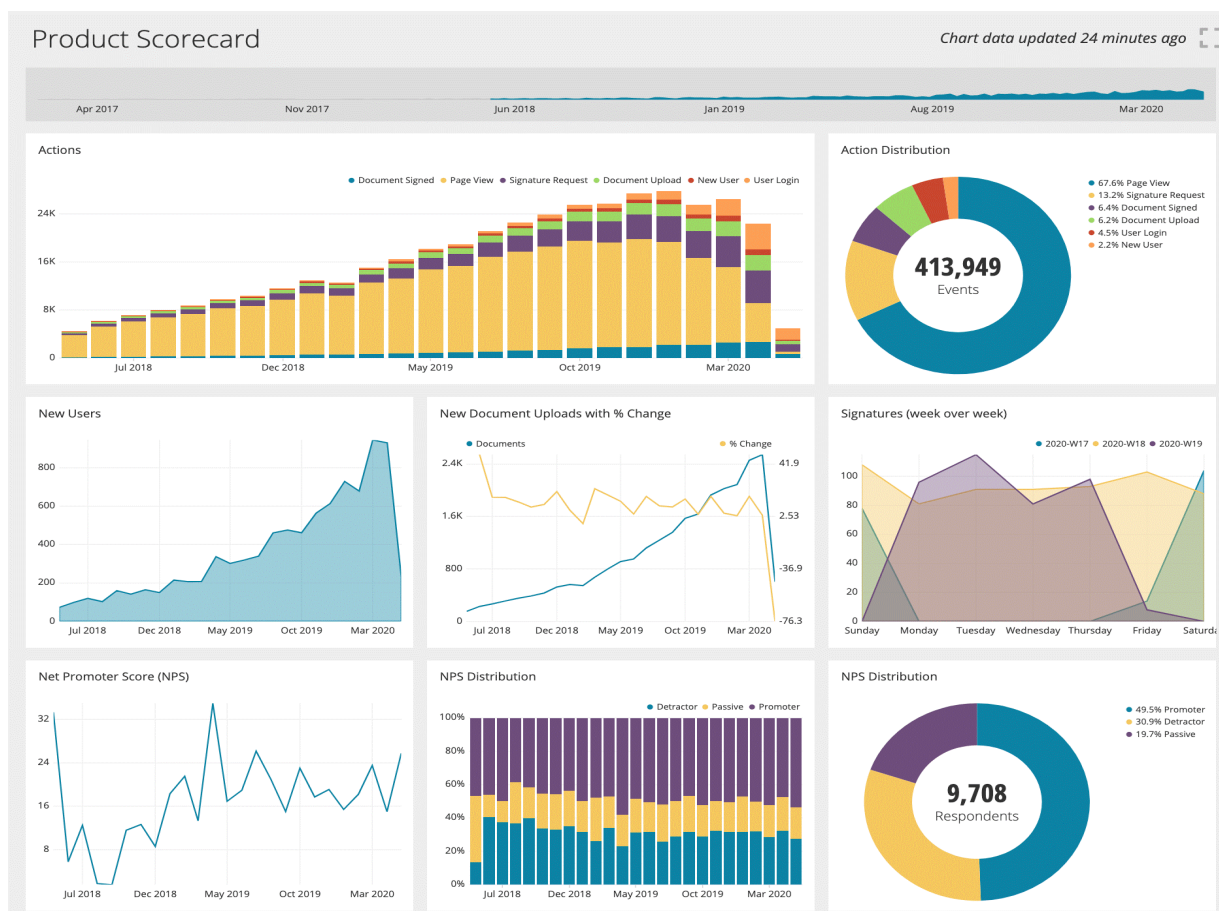


FIGURE 1.4 – Tableau de bord de Chartio.

D. Holistics

L'outil se présente comme étant en libre-service et propose la création de tableaux de bord interactifs en temps réel. Il est également possible d'envoyer des rapports par courrier électronique depuis Holistics. Il coûte moins cher que beaucoup d'autres ; à partir de 50 dollars par mois, il est possible de faire une immersion complète en matière de data-visualisation. C'est un outil efficace pour faire des rapports de données.

Conçu pour les analystes de données, Holistics fournit un espace de travail idéal pour analyser les données. L'analyse intégrée aide à créer des tableaux de bord externes avec lesquels les partenaires/clients peuvent interagir dans le cadre d'une même application.

Néanmoins, pour l'utiliser il faut avoir certaines notions SQL car cette solution fonctionne dans un environnement SQL. Il est aussi possible d'y intégrer plusieurs bases de données SQL [4].

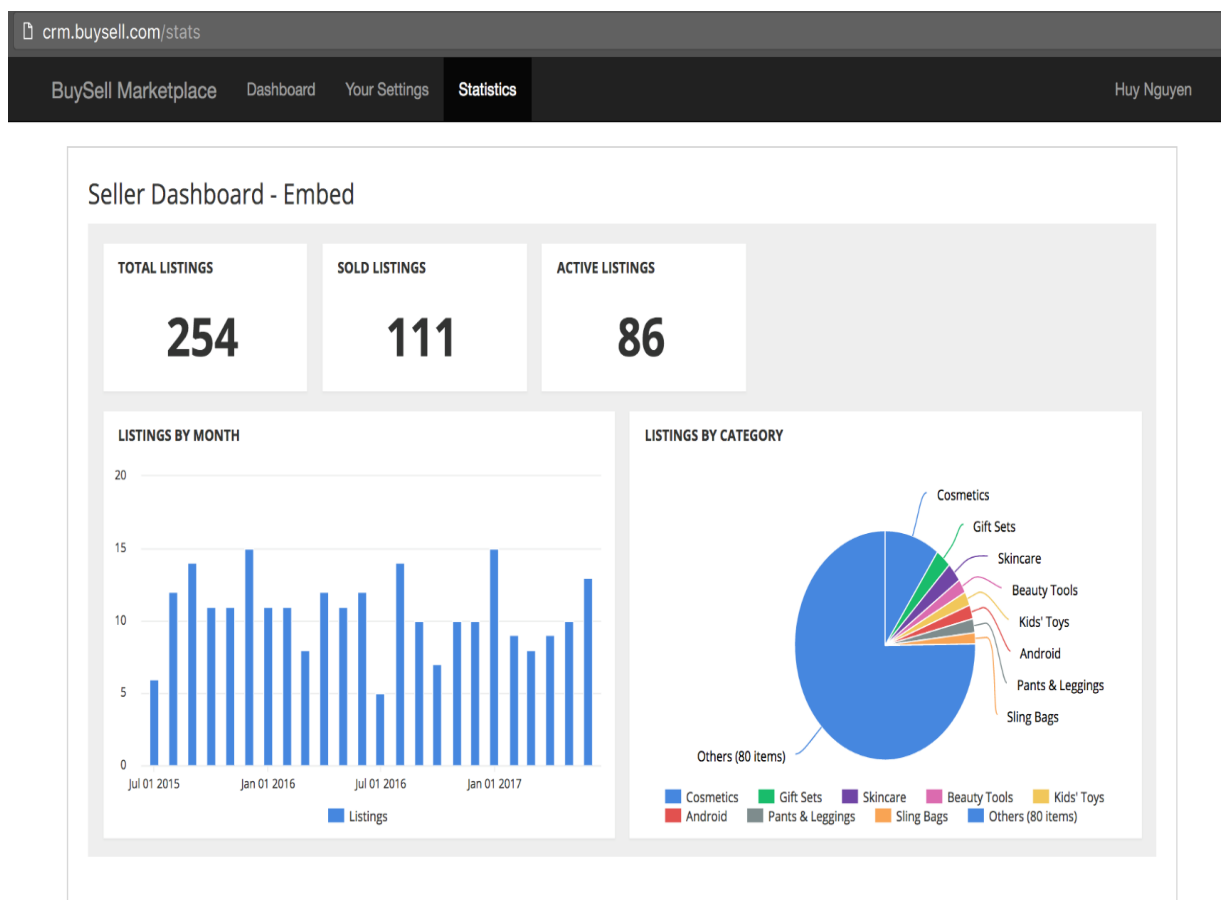


FIGURE 1.5 – Tableau de bord de Holistics.

E. Tableau

Tableau est une référence du marché. C'est clairement un poids lourd. Tableau permet de créer des tableaux, des graphiques, des infographies, des cartes à partir de n'importe quelles données (SQL, Excel...). L'interface est particulièrement intuitive et permet de transférer les données par simple glisser-déposer. Les graphiques sont mis à jour en temps réel et partageables.

Contrairement à tous les produits présentés plus haut, Tableau est disponible en français. C'est vraiment une valeur sûre. Côté tarifs, les plans commencent à 70\$ par utilisateur et par mois [4].

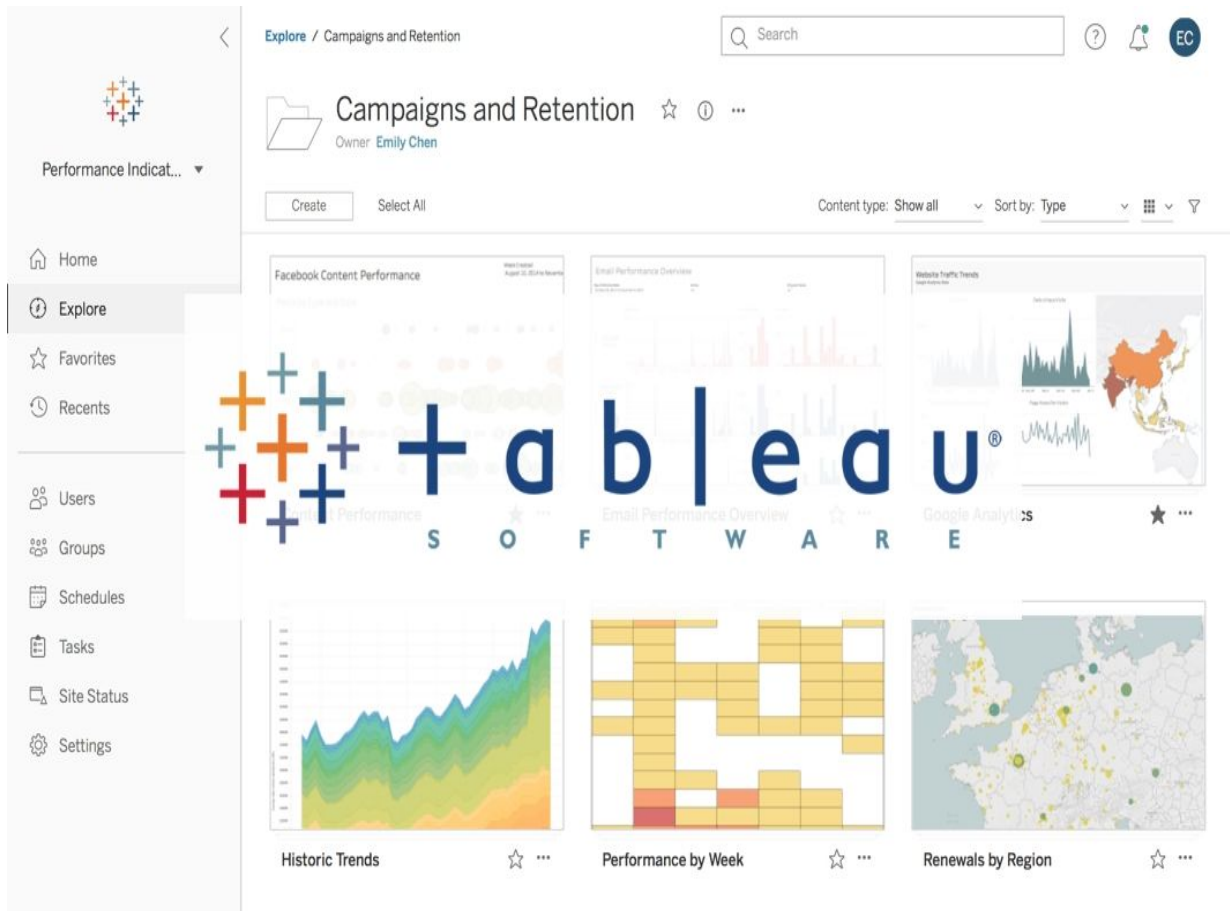


FIGURE 1.6 – Exemple de tableau de bord de Tableau.

F. Klipfolio

D'origine canadienne, Klipfolio permet de créer et de partager des tableaux de bord en temps réel. L'outil propose un nombre très important de connecteurs, plusieurs centaines au total. La solution propose des centaines d'options de visualisation. Klipfolio est payant, à partir de 29\$ par mois, ce qui en fait l'un des meilleurs outils du point de vue du rapport qualité/prix. L'éditeur propose une période d'essai gratuite [5].

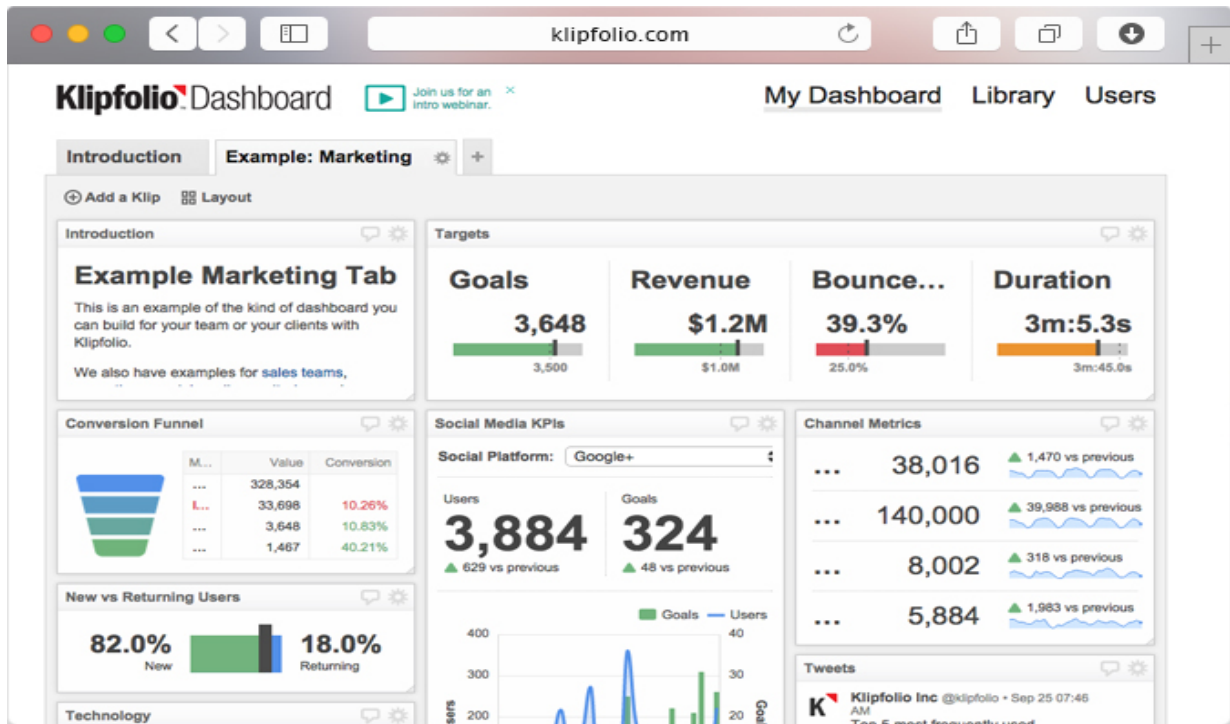


FIGURE 1.7 – Exemple de tableau de bord Klipfolio.

1.3.2 Tableau comparatif entre les outils de data visualisation

L'outil	Sources de données	Les graphiques	Le prix	L'utilisabilité
Chartio	Très varié	Très varié	Trop élevé	Facile
Holistics	Restreintes	Varié	Moyen	Difficile
Tableau	Très varié	Très varié	Moyen	Facile
Klipfolio	Varié	Peu varié	Pas cher	Facile

TABLE 1.1 – Comparaison entre les outils de data visualisation.

1.3.3 Programmation d'un logiciel de visualisation

Les graphiques impressionnant et fait sur mesure nécessitent des compétences en programmation ou une bonne maîtrise de logiciels d'illustration. Autrement dit, dans certains cas il est préférable ou nécessaire de programmer un logiciel de visualisation et ce afin de satisfaire à 100% les besoins vis-à-vis de la visualisation finale, et pour cela il existe de nombreux langages de programmation et outils telle que Angular, Vert.x, Phyton, PHP...

1.4 Enjeux et opportunités de la visualisation des données

Tandis qu'un tableau classique fait apparaître tous les chiffres au même niveau, les rendant difficilement parlants, une simple représentation graphique en nuage de points permet de retranscrire les différences entre les données. Ce qui est invisible dans le tableau prend ainsi forme avec le graphique.

Avec une représentation graphique des données que gère une entreprise, il est possible de mieux comprendre dans quel environnement l'entreprise évolue. En effet, les attentes des prospects, les changements des tendances économiques ou encore le comportement des clients sont identifiés d'une manière plus précise. Il est donc possible de prendre les décisions les plus adéquates afin de rendre ce business plus performant, en accédant à des informations percutantes et plus convaincants auprès des clients et prospects. Et puis il ne faut pas négliger le fait que la data visualisation permet de mieux exploiter les données en notre possession. Ainsi, il est possible d'automatiser différents processus commerciaux et marketing qui utilisent beaucoup de ressources. En plus du gain de temps, cela permet aussi d'investir dans des missions plus ciblées.

Enfin, la data visualisation permet de matérialiser de manière graphique les données chiffrées relatives aux activités, aussi les grandes tendances statistiques sont alors mises en avant et les informations deviennent faciles à comprendre [6].

1.5 Tableaux de bord (Dashboard)

Un tableau de bord de données est un outil de gestion de l'information qui suit, analyse et affiche visuellement des indicateurs de performance clés (KPI), des mesures et des points de données clés pour surveiller la santé d'une entreprise, d'un service ou d'un processus spécifique. Ils sont personnalisables pour répondre aux besoins spécifiques d'un département et d'une entreprise. En coulisse, un tableau de bord se connecte aux fichiers, pièces jointes, services et API, mais en surface affiche toutes ces données sous forme de tableaux, de graphiques en courbes, de graphiques à barres et de jauges. Un tableau de bord de données est le moyen le plus efficace de suivre plusieurs sources de données, car il fournit un emplacement central permettant aux entreprises de surveiller et d'analyser les performances. La surveillance en temps réel réduit les heures d'analyse et la longue ligne de communication qui mettaient auparavant au défi les entreprises [7].

1.6 Mise en forme des Tableaux de bord

Une fois que toutes les données sont disponibles et qu'elles sont en ordre, il est possible de les visualiser. Qu'il s'agisse de faire un rapport, une infographie en ligne ou une représentation artistique de données, il est important de suivre quelques règles élémentaires qui offrent toutes une certaine marge de manœuvre. Dans ce qui suit, nous présenterons les directives qui devront être considérées comme un cadre plutôt qu'un ensemble de règles intangibles. Cependant, il s'agit d'un excellent point de départ pour ce qui débute dans le design de graphiques de données [8].

1.6.1 Les différents types de graphiques

La recherche montre que nous créons 2,5 quintillions d'octets de données chaque jour. Avec autant d'informations accessibles du bout des doigts, il est important de comprendre comment les organiser en informations analysables et exploitables. Pourtant, si on gère plusieurs actifs de contenu avec plusieurs sources de données, il peut être difficile de déterminer comment façonner notre stratégie d'analyse.

C'est là qu'il est utile de connaître les meilleurs types de graphiques de visualisation de données à utiliser.

Bien que tous servent à accélérer et à améliorer l'interprétation des données, tous ne sont pas appropriés pour le même travail. Le choix de la bonne aide visuelle est la clé pour éviter la confusion des utilisateurs et garantir l'exactitude de l'analyse [9].

A. Histogramme

Il s'agit de l'un des types d'outils de visualisation de données les plus courants. Il y a une raison pour laquelle nous apprenons à créer des histogrammes à l'école primaire. C'est un moyen simple et traditionnel de montrer une comparaison entre différents ensembles de données. Il est également possible d'utiliser un histogramme pour suivre les ensembles de données au fil du temps.

Il est facile à lire et à comprendre et puis un ensemble de données peut être modifié sans affecter les autres et aussi la possibilité d'ajouter des étiquettes de données si nécessaire sans trop encombrer le graphique lui-même mais avec trop de catégories, cela peut devenir un peu trop encombré [10].



FIGURE 1.8 – Histogramme

B. Graphique linéaire

C'est un autre de ces types de graphiques standard qui est instantanément reconnaissable. Un graphique linéaire est conçu pour révéler les tendances, les progrès ou les changements qui se produisent au fil du temps. En tant que tel, cela fonctionne mieux lorsque notre ensemble de données est continu plutôt que plein de démarrages et d'arrêts [10].

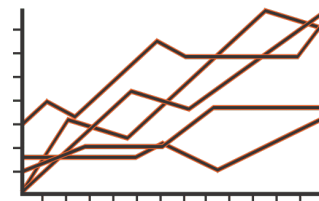


FIGURE 1.9 – Graphique linéaire

C. Graphique à double accès

Alors que la plupart des graphiques de visualisation utilisent un seul axe Y et un seul axe X, un graphique à deux axes incorpore un axe X partagé et deux axes Y séparés. La plupart combinent les fonctionnalités d'un histogramme et d'un graphique linéaire, bien qu'on puisse faire varier les styles graphiques en fonction des données qu'on utilise [10].

Cette disposition permet d'afficher une relation (ou son absence) entre différentes variables, et elle fonctionne mieux lorsqu'on travaille avec trois ensembles de données comme suit :

- Un ensemble de données continues
- Deux ensembles de données regroupés par catégorie

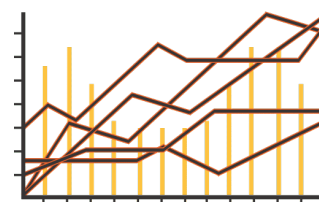


FIGURE 1.10 – Graphique à double accès

D. Graphique à secteurs

Un graphique à secteurs représente un nombre statique, divisé en catégories qui constituent ses parties individuelles. Lorsqu'on utilise un, nous représentons des montants numériques en pourcentages. Lorsqu'on additionne toutes les portions séparées, elles devraient totaliser 100% [10].

Ceux-ci sont particulièrement utiles dans le marketing numérique, car il est possible de les utiliser pour afficher une ventilation de :

- Parts de marché.
- Dépenses de marketing.
- Données démographiques des clients.
- Sources de trafic en ligne.

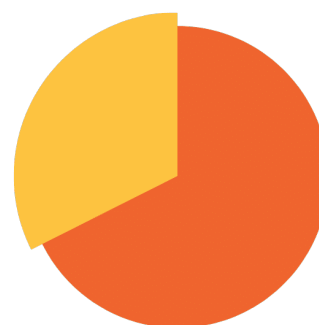


FIGURE 1.11 – Graphique à secteurs

E. Graphique Nuage de points

Ce type de visualisation est également appelé un scattergramme, et il représente différentes variables tracées le long de deux axes. Notez que l'axe X et l'axe Y sont tous deux des axes de valeurs car un nuage de points n'utilise pas d'axe des catégories.

Ce type de visualisation de données fonctionne mieux lorsqu'on souhaite analyser plusieurs points de données et qu'on recherche des similitudes au sein de l'ensemble de données. Ce faisant, on peut remarquer les valeurs aberrantes et mieux comprendre la distribution globale des données [10].

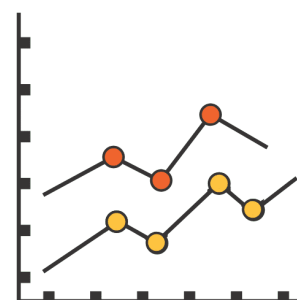


FIGURE 1.12 – Graphique Nuage de points

F. Graphique à bulles

Comme un nuage de points, un graphique à bulles peut également afficher des relations ou une distribution.

Cependant, dans cette variante, on remplace les points de données par des bulles. En variant les tailles de la bulle, il est possible de représenter un troisième ensemble de données.

Comme avec un graphique en nuage de points, un graphique à bulles n'utilise pas d'axe des catégories. Les ensembles de données sont tracés en tant que valeurs X, valeurs Y et valeurs Z (taille de la bulle) [10].



FIGURE 1.13 – Graphique à bulles

G. Graphique à puces

Un graphique à puces peut nous aider à suivre visuellement nos progrès sur un objectif précis. Similaire dans la mise en page à un graphique à barres, ceux-ci intègrent également d'autres éléments visuels. Lorsqu'on utilise un graphique à puces, on commence par une mesure principale, puis comparer cette mesure à une autre (ou plusieurs) mesure pour trouver une signification et un lien plus profonds [10].

1.6.2 La notion du design

Le design d'un graphique obéit à un flux familier. Les données sont récupérées et codées à l'aide de cercles, de barres et de couleurs. À ce stade, le lecteur du graphique doit décoder les conventions adoptées [11].

William Cleveland et Robert McGill ont écrit de façon détaillée sur le codage. Certains codages fonctionnent mieux que d'autres. Mais peu importe celui qu'on adopte si le lecteur ne sait pas ce que le codage représente. S'il est dans l'incapacité de décoder le graphique,

le temps consacré à son design est un temps perdu [12].

De ce fait, il est primordiale de respecter ces notions :

- **Libeller les axes** : dans le même ordre d'idée que l'explication des codes utilisés, il est impératif de toujours libeller les axes. Dans le cas contraire, ceux-ci risquent de n'être là que pour la décoration. Libellez les axes de telle sorte que le lecteur connaisse le type d'échelle tracé.
- **Une géométrie sous contrôle** : les auteurs font souvent une autre erreur, celle de représenter les valeurs à l'aide de formes bidimensionnelles dont ils n'utilisent qu'une seule dimension. Les rectangles d'un graphique en barres sont bidimensionnels, mais seule la longueur sert d'indicateur. La largeur ne représente rien. Cependant, quand on crée un graphique en bulles, on recourt à une surface pour représenter les valeurs. Les débutants utilisent souvent le rayon ou le diamètre, et l'échelle est alors totalement décalée.
- **Inclure les sources** : cela va sans dire, mais beaucoup l'oublie. D'où viennent nos données? Le fait d'asseoir l'honorabilité de notre graphique, la mention des sources permet au lecteur de contrôler les informations ou d'analyser les données. Elle offre aussi un contexte plus large aux chiffres.
- **Respecter les lecteurs du Dashboard** : enfin, il faut toujours prendre en compte l'auditoire et l'objectif du graphique. Par exemple, un graphique destiné à un diaporama doit être simple. On peut inclure une multitude de détails, mais ceux-ci ne seront visibles que par les seules personnes assises au premier rang. En revanche, si on crée un poster qui a pour vocation d'être étudié et examiné, nous pouvons insérer un plus grand nombre de détails.

1.6.3 Data visualisation et infographie

L'infographie est un graphique d'information conçu pas à pas à l'aide d'un logiciel comme Illustrator. Elle n'est pas une simple illustration, car elle se compose de texte et d'images. La définition de l'infographie pourrait donc être : ensemble d'informations mises en scène graphiquement. L'infographie n'est pas forcément composée de chiffres. Elle va présenter et exposer des informations de façon didactique (et parfois ludique) ou montrer un processus. Elle peut également être analytique ou descriptive. Pour que ce soit une infographie, il faut que la mise en scène des éléments apporte une autre dimension et délivre visuellement un message, une information.

Contrairement à l'infographie, une data visualisation est le plus souvent générée automatiquement à l'aide d'un outil ou de code. La dataviz est souvent constituée d'une base de données. On représente graphiquement un jeu de données complexes afin de le rendre compréhensible. Certaines visualisations sont intuitives et immédiatement parlantes, d'autres font appel à des représentations qui peuvent déstabiliser le lecteur par leur design original. La frontière entre l'infographie et la data visualisation peut être difficile à déterminer, surtout lorsque l'on assemble des éléments des deux domaines, qui appartiennent l'un comme l'autre à la famille du design d'informations. Ces combinaisons d'éléments graphiques se retrouvent de plus en plus sous forme de tableaux de bord (dashboard) interactifs [13].

1.7 Conclusion

Il existe beaucoup d'outils de data-visualisation, certains sont gratuits et d'autres, très onéreux. Si nous traitons un volume important de données, nous sommes bien conscient de la nécessité de confier certaines tâches d'analyse à de puissants outils.

La data visualisation n'est ni plus ni moins que le moyen de faciliter le traitement de grande quantité de données par une expérience utilisateur soignée et des éléments visuels facilement identifiables. Les nouveaux usages numériques demandent aujourd'hui de comprendre de manière plus précise les données pour faciliter les prises de décision. C'est dans cette optique que nous souhaitons proposer une solution qui permettra à l'entreprise de visualiser leurs données à travers des différents graphiques déjà cités (graphique à bulle, graphique à secteurs, histogramme ...)

Le chapitre suivant sera consacré à l'analyse des besoins et à la conception de notre application.

Analyse & spécification des besoins

2.1 Introduction

Le développement de n'importe quelle application informatique nécessite de connaître les besoins techniques, ergonomiques et fonctionnels du client. C'est pourquoi nous avons consacré ce chapitre à la présentation l'organisme d'accueil qui est l'Entreprise Tech Instinct ,ces missions et activités. Par la suite, nous allons cerner la problématique ainsi que les motivations assignées dans le but d'assurer une meilleure gestion pour nos données, puis nous passons à la spécification des besoins fonctionnels et non fonctionnels de notre application. Par la suite, nous allons présenté les acteurs qui interagissent avec le système , ensuite nous allons modéliser le tout à travers un diagramme de cas d'utilisation global. Enfin nous terminerons par la description textuelle des cas d'utilisation.

2.2 Présentation de l'organisme d'accueil

2.2.1 Présentation de Tech Instinct

Tech Instinct est une entreprise algérienne, basée à Bejaia, créée en juillet 2018. Composée d'une dizaine d'ingénieurs. Tech Instinct est spécialisée dans le conseil et la réalisation IT, elle accompagne les TPE/PME et les startups pour leur offrir le meilleur conseil en technologies et les aider à créer les solutions adéquates, répondant à leurs besoins, le tout avec des budgets maîtrisés. L'expérience de ses consultants, certains ayant évolué dans les plus grandes entreprises françaises, formés aux dernières technologies et aux pratiques agiles permet de créer des produits fiables et d'excellente qualité. Tech Instinct propose des prestations de qualité, au moins équivalentes aux prestations On Shore (en France et en Europe) et ce à des tarifs très compétitifs [14].



FIGURE 2.1 – Logo entreprise « Tech Instinct ».

2.2.2 Domaines d’expertise et technologies adoptées

Tech instinct dispose d’une forte expertise dans les architectures SI, le développement de plateformes SAAS, développement d’API, d’applications web et mobiles, développement Fullstack, Devops et industrialisation et Cloud (AWS / GCP). Pour la réalisation de ses projets, elle repose sur un ensemble de technologies tel que : ANGULAR, PYTHON, HTML 5, MYSQL, CSS, BOOTSTRAP, PHP, JAVA, CASSANDRA et MANGODB, etc [14].

2.3 Problématique et motivation

Aujourd’hui, la Data Visualization est devenue incontournable. Utilisée pour communiquer, illustrer et résumer, nous voyons quotidiennement passer sous nos yeux des dizaines de visuels : graphiques, diagrammes, nuages de points, cartes géographiques Et ce dans différents contextes : les sciences, l’économie, le journalisme, les réseaux sociaux ...

Il existe pléthore d’applications et de logiciels permettant de créer des graphiques et des visuels à partir d’un ensemble de données. Ceci est facilité par la disponibilité de bibliothèques facilitant l’écriture de ce type d’applications. Le web n’étant pas en reste, une infinité de bibliothèques existent pour faire de la Data Visualization.

Les applications modernes, qu’elles soient des plateformes, des logiciels de gestion, des outils de monitoring, des réseaux sociaux ... intègrent la Data Visualization. En effet, cela constitue un moyen très efficace de représenter et de présenter l’information.

Implémenter la Data Visualization des applications peut s'avérer complexe pour plusieurs raisons, notamment :

1. Le coût du développement
2. La maîtrise des technologies de DataViz
3. La maintenance suite aux évolutions rapides des bibliothèques
4. Le manque d'agilité

C'est pour ces raisons que nous avons décidé de concevoir et réalisé une plateforme qui résout ces différents problèmes.

2.4 Besoins de l'entreprise d'accueil

Avant de passer à la phase pratique et de réalisation de notre application, la bonne pratique nécessite de définir un cahier de charges détaillé sur le projet.

2.4.1 Besoins fonctionnels

Les besoins fonctionnels représentent les fonctionnalités que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Les besoins fonctionnels principaux sont résumés ci-dessous :

- Automatiser le processus de gestion des données.
- L'authentification : une fois accédé à l'application, l'utilisateur doit introduire son login et mot de passe pour s'authentifier et avoir ensuite accès à d'autres fonctionnalités.
- Générer des graphes pour faciliter la tâche aux utilisateurs. Entre autres, ces fonctionnalités permettent d'avoir une vue globale sur les données des graphes.
- Gérer leurs projets, les manifestes et les sauvegarder dans la base de données.
- Gérer les Dashboard et ses graphes.
- Mettre en place un mécanisme permettant d'exporter des documents sous format PDF ou PPT.
- Consulter et rechercher à tout moment les informations afin de mettre à jour les données, projets et Dashboard.

2.4.2 Besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou de type de conception. La liste suivante représente les besoins non-fonctionnels de notre application :

- Contraintes d'implémentation (langage de programmation, type de SGBD, SE).
- Performance : l'application doit assurer un temps de réponse minime, tout en répondant aux exigences de l'utilisateur.
- Sécurité : les données recueillies par l'application ne doivent être accessibles que par le personnel autorisé.
- Modularité du code : le code doit être lisible afin de simplifier la maintenance.
- Ergonomie : l'interface doit être simple à utiliser, les informations doivent être lisibles et ce avec un minimum d'effort.
- Coût

2.5 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données. En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. D'autre part, le même acteur peut jouer alternativement plusieurs rôles et le même rôle peut être joué par plusieurs acteurs [15]. Dans notre cas, nous avons identifié trois acteurs :

1. **Visiteur** : il aura comme tâches

- Consulter la vidéo démonstrative.
- Créer son compte utilisateur.

2. Utilisateur : il aura comme taches

- Gestion des projets (Ajouter, Modifier, Supprimer, Rechercher)
- Gestion des Dashboard (Ajouter, Modifier, Supprimer, Rechercher)
- Gestion des graphes (Ajouter, Modifier, Supprimer)
- En plus de ces fonctionnalités, l'utilisateur peut faire les fonctionnalités du visiteur.

3. Administrateur : il aura comme taches

- Gestion des comptes utilisateurs (Ajouter un compte, Modifier un compte, Supprimer un compte et Rechercher un compte).
- En plus de ces fonctionnalités, l'administrateur peut faire les fonctionnalités de l'utilisateur et du visiteur.

2.6 Identification des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il modélise un service rendu par le système, exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné [15]. Lors de notre étude, nous avons pu identifier les actions importantes pour chaque acteur qui sont illustrées dans le Tableau 3.1 :

N°	Cas D'utilisation		Acteurs
1	Authentification		Utilisateur , Administrateur
2	Consulter la vidéo démonstrative		Tous les acteurs
3	Gestion des projets	Rechercher un projet	Utilisateur, Administrateur
		Créer un projet	Utilisateur, Administrateur

		Modifier un projet	Utilisateur, Administrateur
		Supprimer un projet	Utilisateur, Administrateur
4	Gestion des Dashboards	Rechercher un Dashboard	Utilisateur, Administrateur
		Créer un Dashboard	Utilisateur, Administrateur
		Modifier un Dashboard	Utilisateur, Administrateur
		Supprimer un Dashboard	Utilisateur, Administrateur
5	Gestion des graphes d'un Dashboard	Créer un graphe	Utilisateur, Administrateur
		Modifier un graphe	Utilisateur, Administrateur
		Supprimer un graphe	Utilisateur, Administrateur
6	Gestion des comptes	Rechercher un compte	Administrateur
		Créer un compte	Administrateur, Visiteur
		Modifier un compte	Administrateur
		Supprimer un compte	Administrateur

TABLE 2.1 – Identification des cas d'utilisation.

2.6.1 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation servent à décrire un ensemble d'actions (cas d'utilisation) que certains systèmes devraient ou peuvent exécuter en collaboration avec un ou plusieurs utilisateurs externes du système (acteurs). Chaque cas d'utilisation doit fournir des résultats observables et utiles aux acteurs ou autres parties prenantes du système. La Figure 3.1 représente le diagramme de cas d'utilisation global qui décrit l'ensemble des cas d'utilisation associés aux différents acteurs de notre système :

CHAPITRE 2. ANALYSE & SPÉCIFICATION DES BESOINS

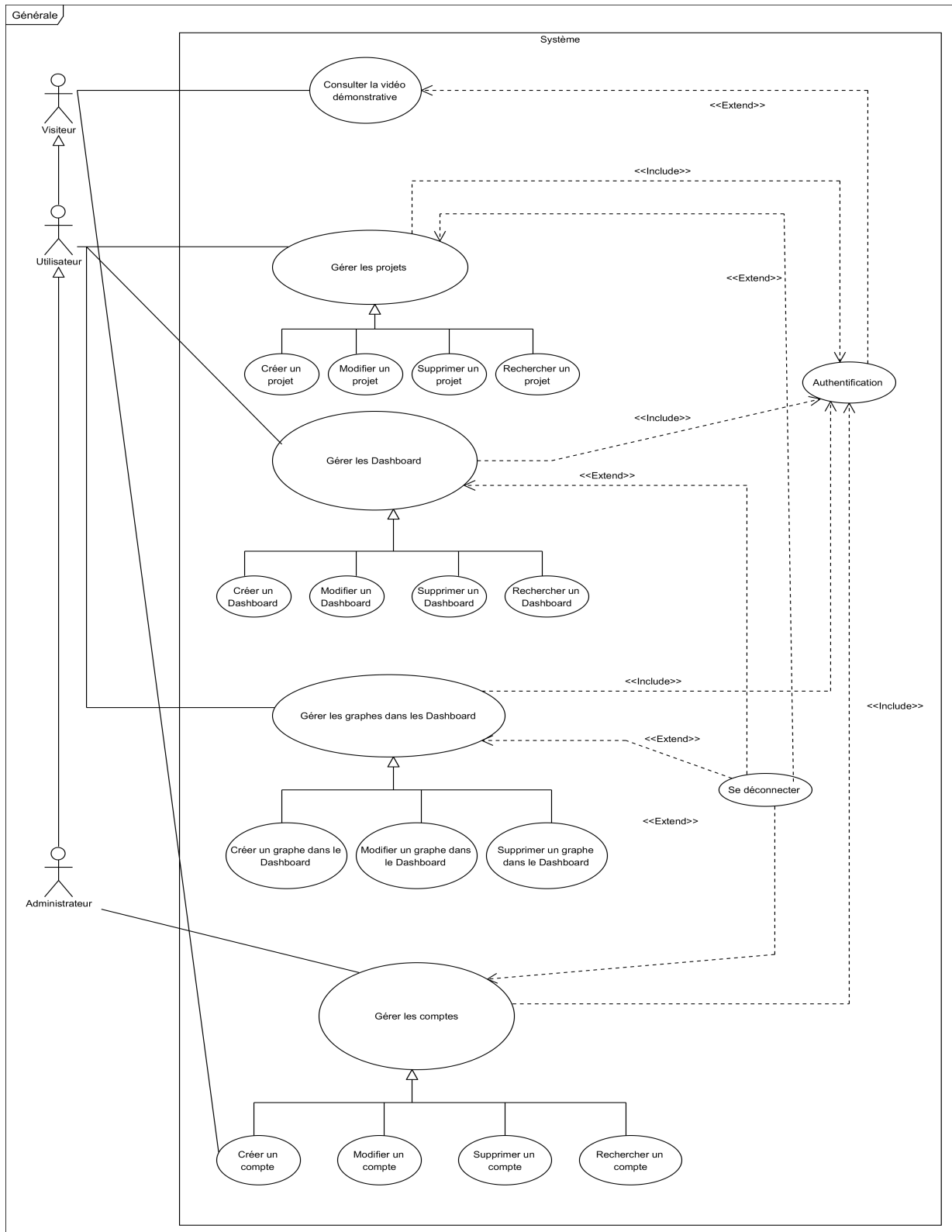


FIGURE 2.2 – Diagramme de cas d'utilisation globale du système.

2.6.2 Description textuelle des cas d'utilisation

Dans cette section, nous présentons une description détaillée de chaque cas d'utilisation. Il s'agit de définir les scénarios possibles

- **Cas d'utilisation « Authentification »**

Le tableau suivant présente une description du cas d'utilisation authentification :

Sommaire d'identification	
Titre	Authentification.
Résumé	Ce cas d'utilisation permet à l'utilisateur d'accéder à son compte. Le système va ensuite gérer les droits d'accès et d'affichage.
Acteur	Utilisateur, Administrateur.
Description des scénarios	
Préconditions	Démarrage de l'application.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur introduit son login et mot de passe. 2. Le système vérifie l'existence du compte. 3. Le système affiche l'interface principale selon les droits de l'utilisateur
Enchaînements d'erreurs	Le système affiche un message d'erreur si le compte n'existe pas ou le mot de passe/login est incorrect et réaffiche le formulaire d'authentification.
Post conditions	L'utilisateur accède à la partie du système qui lui est réservée.

TABLE 2.2 – Description du cas d'utilisation « Authentification ».

Le diagramme de cas d'utilisation "Authentification" est présenté dans la figure suivante :

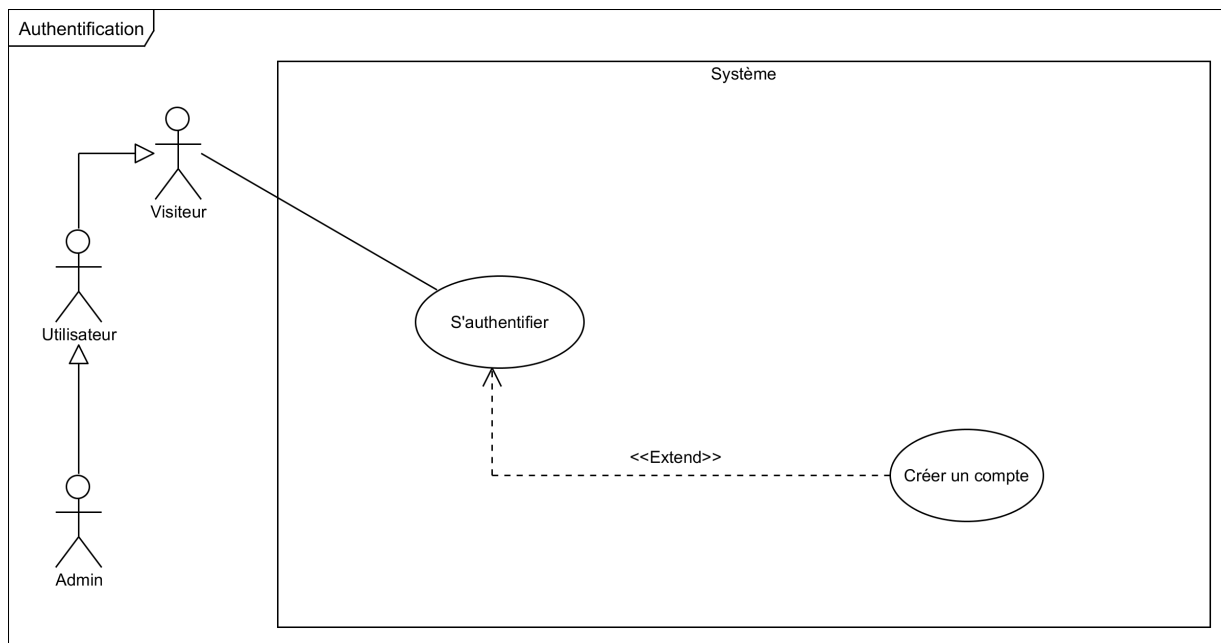


FIGURE 2.3 – Diagramme de cas d'utilisation « Authentification ».

● **Cas d'utilisation « Consultation de la vidéo démonstrative»**

Le tableau suivant présente une description du cas d'utilisation " Consultation de la vidéo démonstrative " :

Sommaire d'identification	
Titre	Consultation de la vidéo démonstrative.
Résumé	Ce cas d'utilisation permet à l'utilisateur de consulter la vidéo démonstrative de l'application.
Acteur	Tous les acteurs.
Description des scénarios	
Préconditions	Démarrage de l'application.

Scénario nominal	Cas : Voir la vidéo démonstrative <ul style="list-style-type: none"> Le système lui affiche la vidéo démonstrative de l'application.
Enchaînements d'erreurs	Le système affiche un message d'erreur si l'utilisateur essaye d'accéder a une zone non autorisée.
Post conditions	L'utilisateur accède à la partie du système qui lui est réservée.

TABLE 2.3 – Description du cas d'utilisation «Consultation de la vidéo démonstrative».

Le diagramme de cas d'utilisation "Consultation de la vidéo démonstrative" est présenté dans la figure suivante :

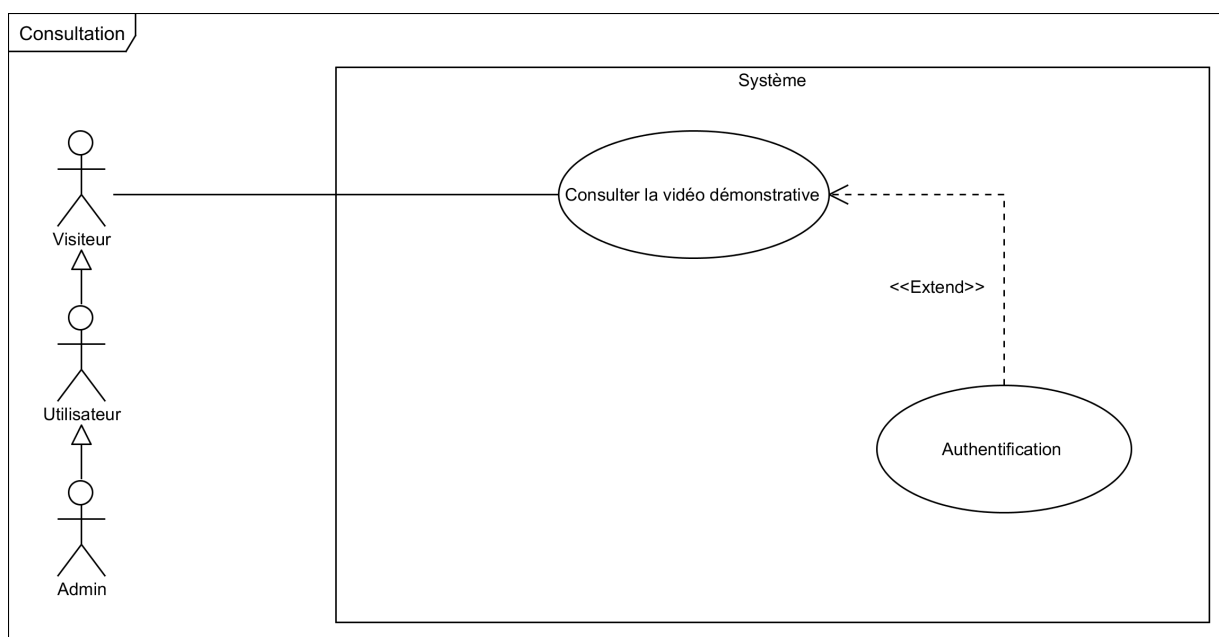


FIGURE 2.4 – Diagramme de cas d'utilisation «Consultation de la vidéo démonstrative».

● **Cas d'utilisation « Gestion des projets »**

Dans le tableau suivant, nous présentons une description du cas d'utilisation " Gestion des projets " :

Sommaire d'identification	
Titre	Gestion des projets.
Résumé	Ce cas d'utilisation permet de créer, modifier, supprimer ou rechercher un projet.
Acteur	Utilisateur, Administrateur
Description des scénarios	
Préconditions	<ol style="list-style-type: none"> 1. Démarrage de l'application 2. L'utilisateur authentifié
Scénario nominal	<p>Cas 1 : Créer un projet</p> <ul style="list-style-type: none"> • L'utilisateur choisit d'ajouter un projet. • Le système affiche le formulaire à remplir. • L'utilisateur remplit et valide le formulaire. • Le système ajoute les informations dans la base de données. • Le système met à jour la liste des projets. <p>Cas 2 : Modifier un projet</p> <ul style="list-style-type: none"> • L'utilisateur choisit le projet à modifier. • Le système affiche le formulaire des données du projet choisi. • L'utilisateur modifie les champs voulus. • Le système met à jour les informations dans la base de données. • Le système met à jour la liste des projets. <p>Cas 3 : Supprimer un projet</p> <ul style="list-style-type: none"> • L'utilisateur choisit le projet à supprimer. • Le système attend une confirmation. • L'utilisateur confirme ou annule la suppression.

	<ul style="list-style-type: none"> • Si l'utilisateur confirme la suppression, le système supprime le projet de la base de données. • Le système met à jour la liste des projets. <p>Cas 4 : Rechercher un projet</p> <ul style="list-style-type: none"> • L'utilisateur saisie le nom du projet qu'il veut rechercher dans la barre de recherche. • Le système lui affiche le projet.
Enchaînements d'erreurs	<ol style="list-style-type: none"> 1. Si les informations introduites sont incorrectes ou non valides, le système affiche un message d'erreur et réaffiche la fenêtre concernée. 2. Création, Modification avec des champs vides ou champs non conforme aux types : un message d'erreur sera affiché. 3. Recherche : projet non trouvé, un message d'erreur sera affiché.
Post conditions	Le système met à jour la base de données.

TABLE 2.4 – Description du cas d'utilisation « Gestion des projets ».

Le diagramme de cas d'utilisation "Gestion des projets" est présenté dans la figure suivante :

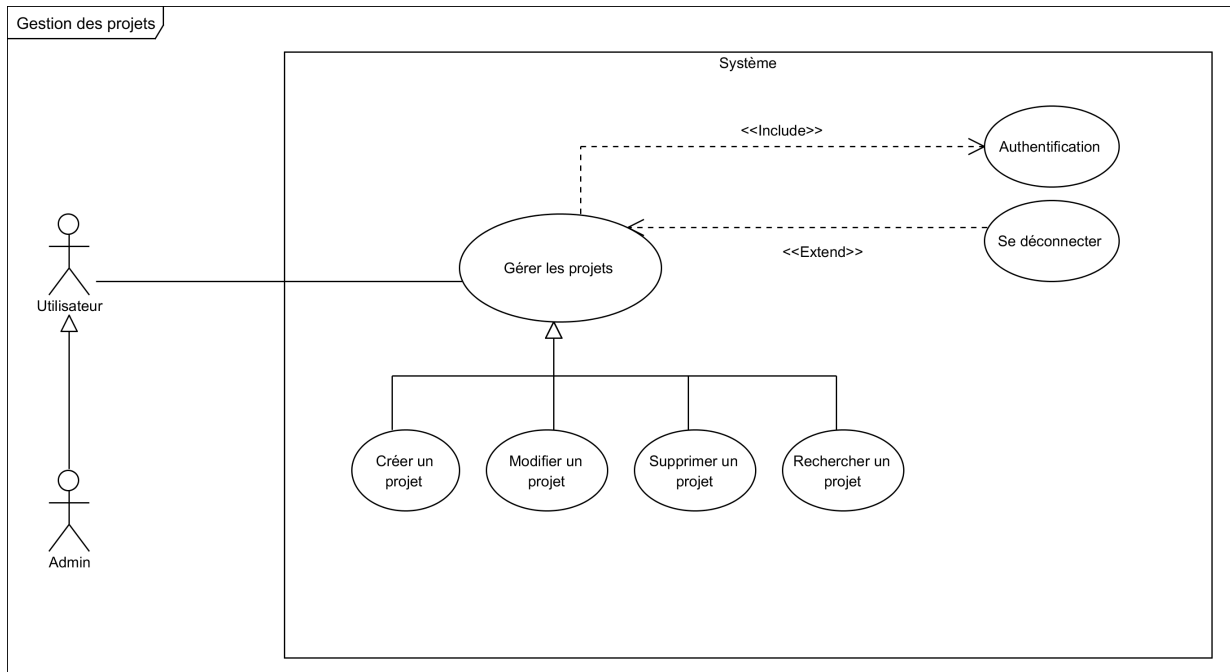


FIGURE 2.5 – Diagramme de cas d'utilisation « Gestion des projets ».

● Cas d'utilisation « Gestion des Dashboards »

Dans le tableau suivant, nous présentons une description du cas d'utilisation " Gestion des Dashboards " :

Sommaire d'identification	
Titre	Gestion des Dashboards.
Résumé	Ce cas d'utilisation permet de créer, modifier, supprimer ou rechercher un Dashboards.
Acteur	Utilisateur, Administrateur
Description des scénarios	
Préconditions	1. Démarrage de l'application 2. L'utilisateur authentifié

<p>Scénario nominal</p>	<p>Cas 1 : Créer un Dashboard</p> <ul style="list-style-type: none">• L'utilisateur choisit d'ajouter un Dashboard.• Le système affiche le formulaire à remplir.• L'utilisateur remplit et valide le formulaire.• Le système ajoute les informations dans la base de données.• Le système met à jour la liste des Dashboard. <p>Cas 2 : Modifier un Dashboard</p> <ul style="list-style-type: none">• L'utilisateur choisit le Dashboard à modifier.• Le système affiche les données à modifier du Dashboard choisi.• L'utilisateur modifie les données voulues.• Le système met à jour les informations dans la base de données.• Le système met à jour la liste des Dashboard. <p>Cas 3 : Supprimer un Dashboard</p> <ul style="list-style-type: none">• L'utilisateur choisit le Dashboard à supprimer.• Le système attend une confirmation.• L'utilisateur confirme ou annule la suppression.• Si l'utilisateur confirme la suppression, le système supprime le Dashboard de la base de données.• Le système met à jour la liste des Dashboard. <p>Cas 4 : Rechercher un Dashboard</p> <ul style="list-style-type: none">• L'utilisateur saisit le Dashboard qu'il veut rechercher dans la barre de recherche.• Le système lui affiche le Dashboard.
-------------------------	---

Enchaînements d'erreurs	<ol style="list-style-type: none"> 1. Si les informations introduites sont incorrectes le système affiche un message d'erreur et réaffiche la fenêtre concernée. 2. Création, Modification avec des champs vides ou champs non conforme aux types : un message d'erreur sera affiché. 3. Recherche : Dashboard non trouvé, un message d'erreur sera affiché.
Post conditions	Le système met à jour la base de données.

TABLE 2.5 – Description du cas d'utilisation « Gestion des Dashboard ».

Le diagramme de cas d'utilisation "Gestion des Dashboard" est présenté dans la figure suivante :

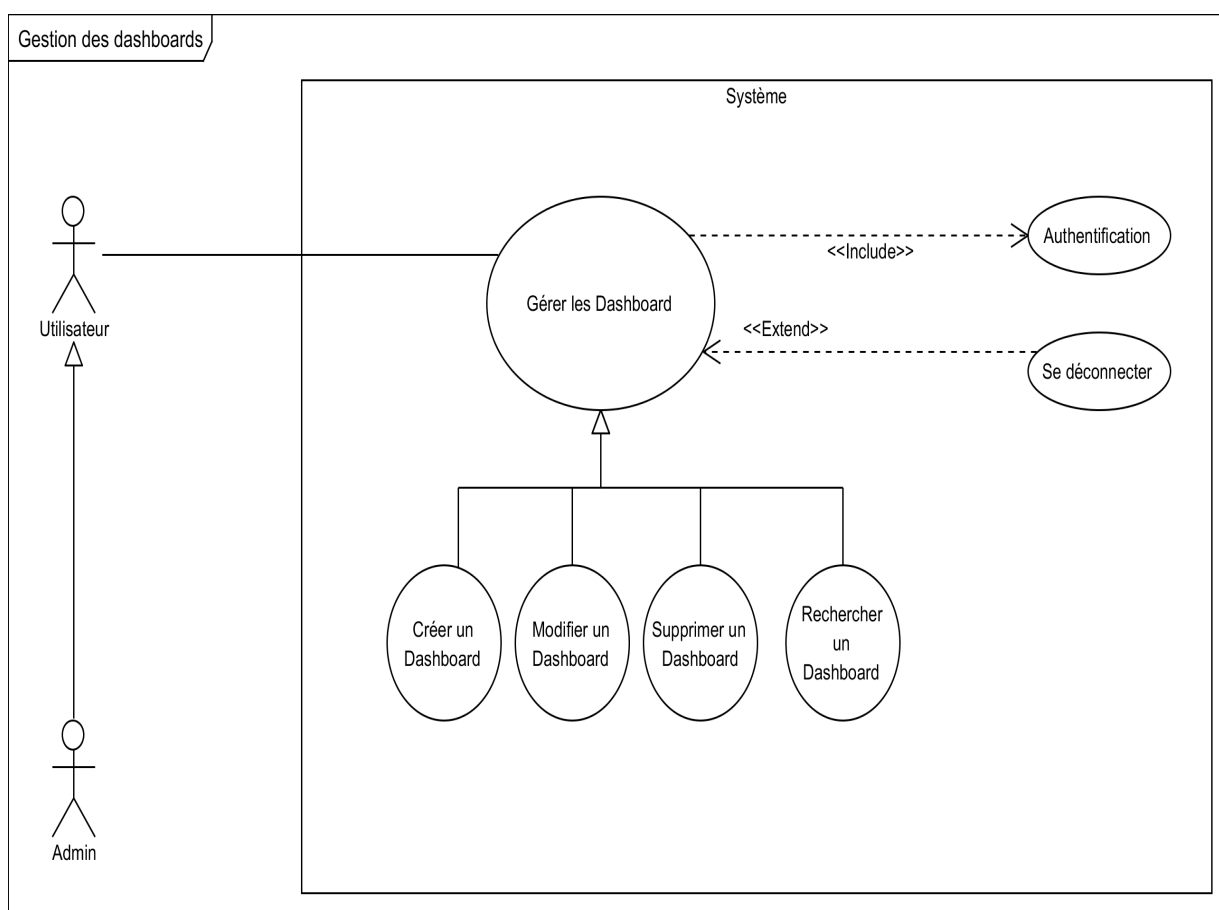


FIGURE 2.6 – Diagramme de cas d'utilisation « Gestion des Dashboard ».

● **Cas d'utilisation « Gestion des graphes dans les Dashboard »**

Dans le tableau suivant, nous présentons une description du cas d'utilisation " Gestion des graphes dans les Dashboards " :

Sommaire d'identification	
Titre	Gestion des graphes dans les Dashboard.
Résumé	Ce cas d'utilisation permet de créer, modifier, supprimer un graphe dans un Dashboard.
Acteur	Utilisateur, Administrateur
Description des scénarios	
Préconditions	<ol style="list-style-type: none"> 1. Démarrage de l'application 2. L'utilisateur authentifié 3. L'utilisateur doit sélectionner un Dashboard
Scénario nominal	<p>Cas 1 : Créer un graphe</p> <ul style="list-style-type: none"> ● L'utilisateur sélectionne le graphe à ajouter. ● Le système affiche la fenêtre de manipulation des données. ● L'utilisateur choisit le fichier à utiliser dans ce graphe (Un seul fichier par graphe mais peut avoir plusieurs fichiers par Dashboard). L'utilisateur peut sélectionner un fichier déjà importé, ou importer un nouveau fichier (Le fichier doit être du type csv xlsx). ● Le système affiche l'aperçus du fichier importé. ● L'utilisateur adapte les données / les colonnes au graphe sélectionné. ● Le système met à jour le graphe selon l'adaptation des données. ● L'utilisateur clique pour ajouter le graphe. ● Le système met à jour le Dashboard.

	<p>Cas 2 : Modifier un graphe</p> <ul style="list-style-type: none"> • L'utilisateur clique pour modifier un graphe sélectionné. • Le système affiche la fenêtre de manipulation des données. • L'utilisateur peut changer le fichier à utiliser dans ce graphe ou changer le type du graphe ou le nom du graphe. • L'utilisateur clique pour confirmer les modification. • Le système met à jour l'aperçus. <p>Cas 3 : Supprimer un graphe</p> <ul style="list-style-type: none"> • L'utilisateur choisit le graphe à supprimer. • Le système attend une confirmation. • L'utilisateur confirme ou annule la suppression. • Si l'utilisateur confirme, le système supprime le graphe du Dashboard. • Le système met à jour le Dashboard.
<p>Enchaînements d'erreurs</p>	<p>1. Si les informations introduites sont incorrectes le système affiche un message d'erreur et réaffiche la fenêtre concernée.</p> <p>2. Création, Modification avec des champs vides ou champs non conforme aux types : un message d'erreur sera affiché.</p>
<p>Post conditions</p>	<p>Le système met à jour la base de données.</p>

TABLE 2.6 – Description du cas d'utilisation « Gestion des graphes dans les Dashboard ».

Le diagramme de cas d'utilisation "Gestion des graphes dans les Dashboard" est présenté dans la figure suivante :

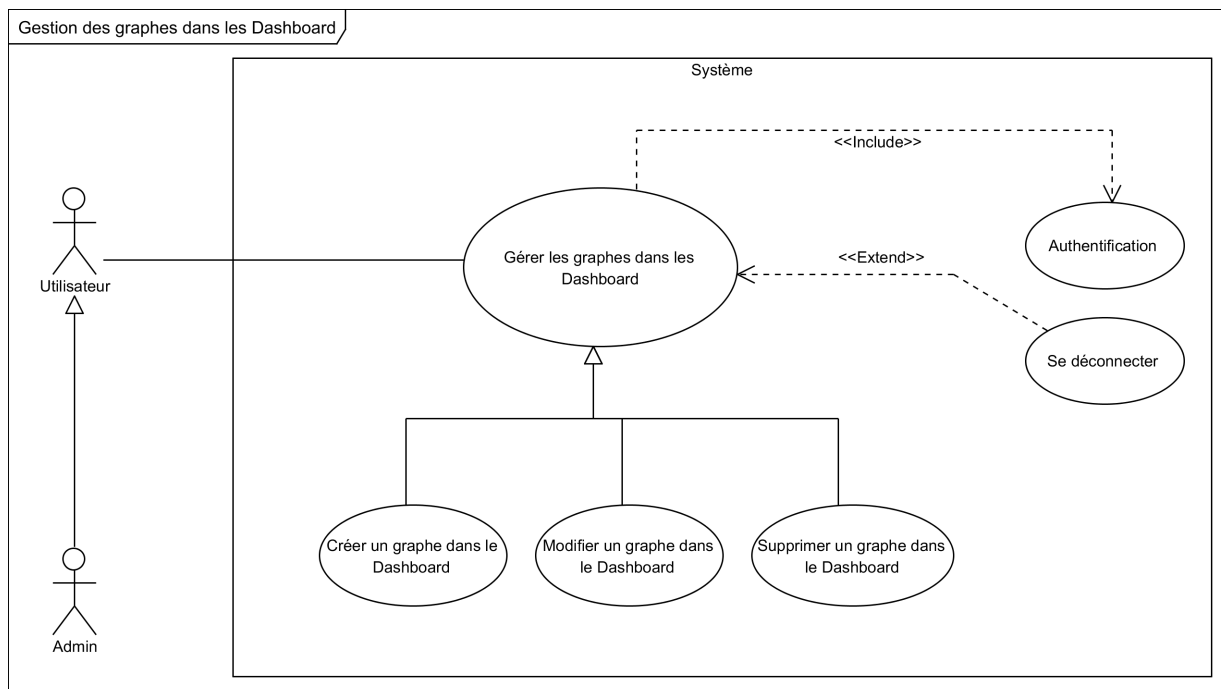


FIGURE 2.7 – Diagramme de cas d'utilisation « Gestion des graphes dans les Dashboard ».

● **Cas d'utilisation « Gestion des comptes »**

Le tableau suivant présente une description du cas d'utilisation "Gestion des comptes" :

Sommaire d'identification	
Titre	Gestion des comptes.
Résumé	Ce cas d'utilisation permet de Créer, Modifier, Supprimer ou Rechercher un compte utilisateur.
Acteur	Administrateur
Description des scénarios	
Préconditions	<ol style="list-style-type: none"> 1. Démarrage de l'application 2. L'utilisateur authentifié en tant que Administrateur

<p>Scénario nominal</p>	<p>Cas 1 : Créer un compte</p> <ul style="list-style-type: none"> • L'administrateur choisit d'ajouter un compte. • Le système affiche le formulaire à remplir. • L'administrateur remplit et valide le formulaire. • Le système ajoute les informations dans la base de données. • Le système met à jour la liste des comptes. <p>Cas 2 : Modifier un compte</p> <ul style="list-style-type: none"> • L'administrateur choisit le compte à modifier. • Le système affiche le formulaire à modifier du compte choisi. • L'administrateur modifie les données voulues. • Le système met à jour les informations dans la base de données. • Le système met à jour la liste des comptes. <p>Cas 3 : Supprimer un compte</p> <ul style="list-style-type: none"> • L'administrateur choisit le compte à supprimer. • Le système attend une confirmation. • L'administrateur confirme ou annule la suppression. • Si l'administrateur confirme la suppression, le système supprime le compte de la base de données. • Le système met à jour la liste des comptes. <p>Cas 4 : Rechercher un compte</p> <ul style="list-style-type: none"> • L'administrateur saisit le compte qu'il veut rechercher dans la barre de recherche. • Le système lui affiche le compte.
<p>Enchaînements d'erreurs</p>	<ol style="list-style-type: none"> 1. Si les informations introduites sont incorrectes le système affiche un message d'erreur et réaffiche la fenêtre concernée. 2. Création, Modification avec des champs vides ou champs non conforme aux types : un message d'erreur sera affiché. 3. Recherche : Compte non trouvé, un message d'erreur sera affiché.
<p>Post conditions</p>	<p>Le système met à jour la base de données.</p>

Le diagramme de cas d'utilisation "Gestion des comptes" est présenté dans la figure suivante :

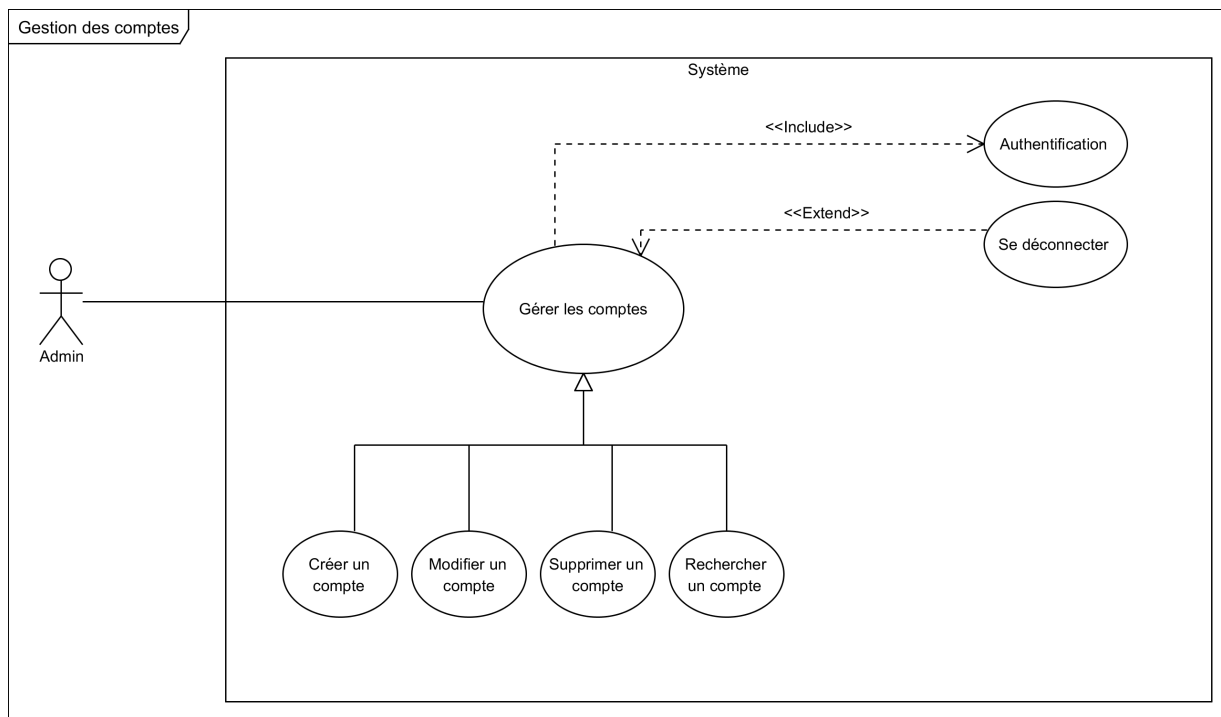


FIGURE 2.8 – Diagramme de cas d'utilisation « Gestion des comptes ».

2.7 Conclusion

Dans ce chapitre nous avons présenté une spécification complète des besoins de notre application en montrant le principe de fonctionnement et les différentes interactions entre les acteurs et le système. Tout d'abord, nous avons présenté l'entreprise Tech Instinct au sein de laquelle nous avons effectué notre stage ainsi que la problématique et les motivations et les besoins fonctionnelles et non fonctionnelles. Ensuite, nous avons identifié les cas d'utilisations et effectué une description textuelle de ces derniers, ce qui va nous permettre d'appréhender plus efficacement le chapitre suivant qui concerne la conception de notre application.

Conception

3.1 Introduction

Après l'étape de l'analyse et spécification des besoins, nous entamons maintenant la phase de conception. Afin d'analyser la dynamique des besoins capturés nous allons commencer par illustrer les différents diagrammes de séquences système représentant les interactions chronologiques entre les acteurs et le système, par la suite nous détaillerons la conception de notre application au moyen du diagramme de classes et du modèle relationnel et nous terminerons par le dictionnaire de données.

3.2 Diagramme de séquence système

Le diagramme de séquence est un diagramme d'interaction UML. Il représente des échanges de messages entre éléments au cours d'une interaction, Il comprend un groupe d'objets, représentés par des lignes de vie. L'intérêt d'un diagramme de séquence système est de décrire textuellement de façon détaillée chaque cas d'utilisation [15]. Afin de mieux comprendre le fonctionnement de notre système nous allons dans ce qui suit détailler les cas d'utilisation présentés dans le chapitre précédent par des diagrammes de séquences système.

3.2.1 Diagramme de séquence Authentification

À chaque authentification, l'utilisateur doit saisir son login et mot de passe. L'utilisateur valide sa demande de connexion, deux cas peuvent se présenter : si les champs sont vides, le système affiche un message d'erreur "Veuillez renseigner tous les champs", si les champs sont bien remplis, le système vérifie l'existence de ce compte. Deux autres cas peuvent se

présenter : si le compte n'existe pas ,affichage d'un message d'erreur 'Ce compte n'existe pas' sinon affichage de l'interface correspondante au statut de l'utilisateur.

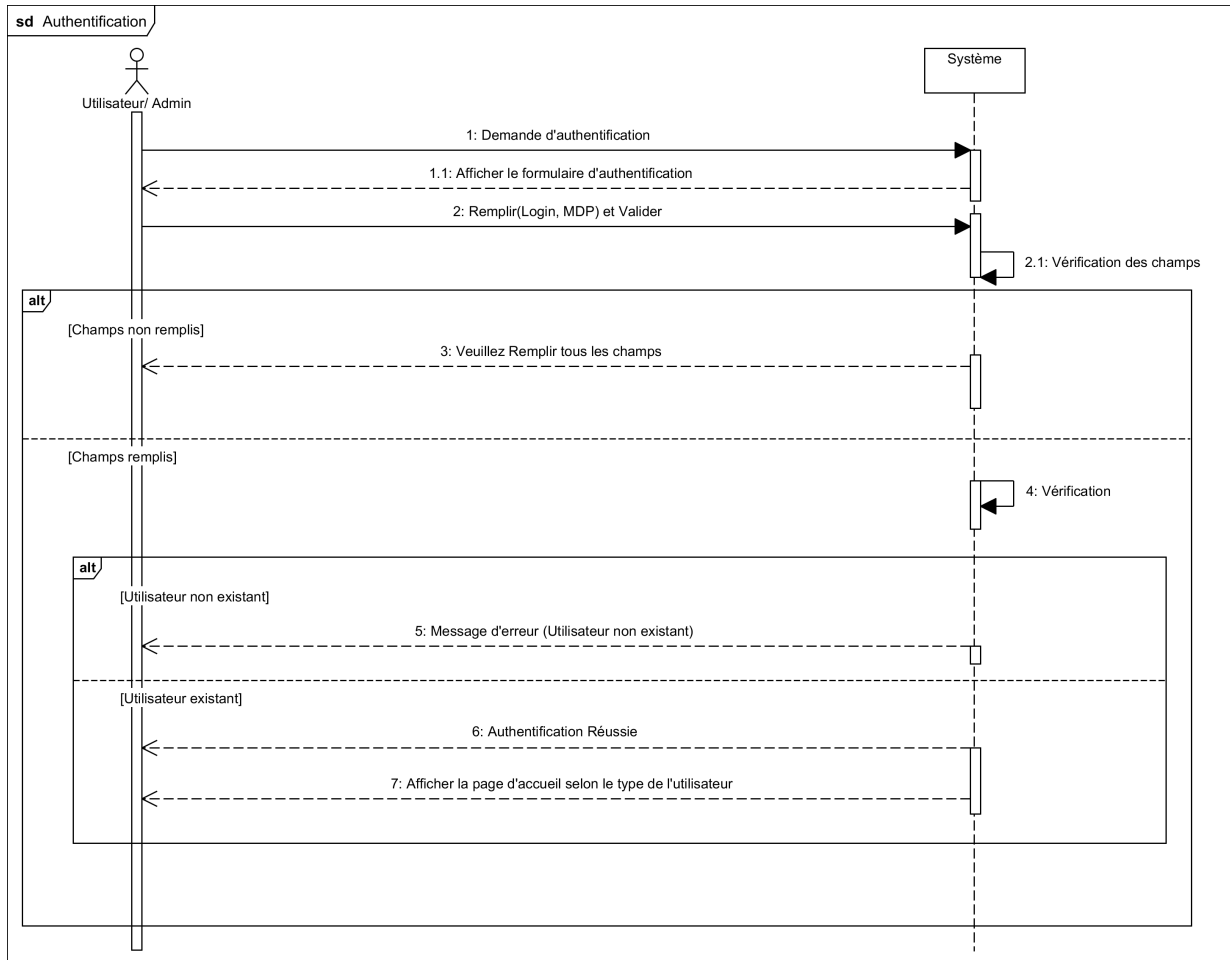


FIGURE 3.1 – Diagramme de séquence Authentification.

3.2.2 Diagramme de séquence Gestion des projets

Après s'être authentifié, l'utilisateur accède à l'interface des projets, ou il peut effectuer les tâches suivantes : créer un projet, modifier un projet, supprimer un projet ou rechercher un projet.

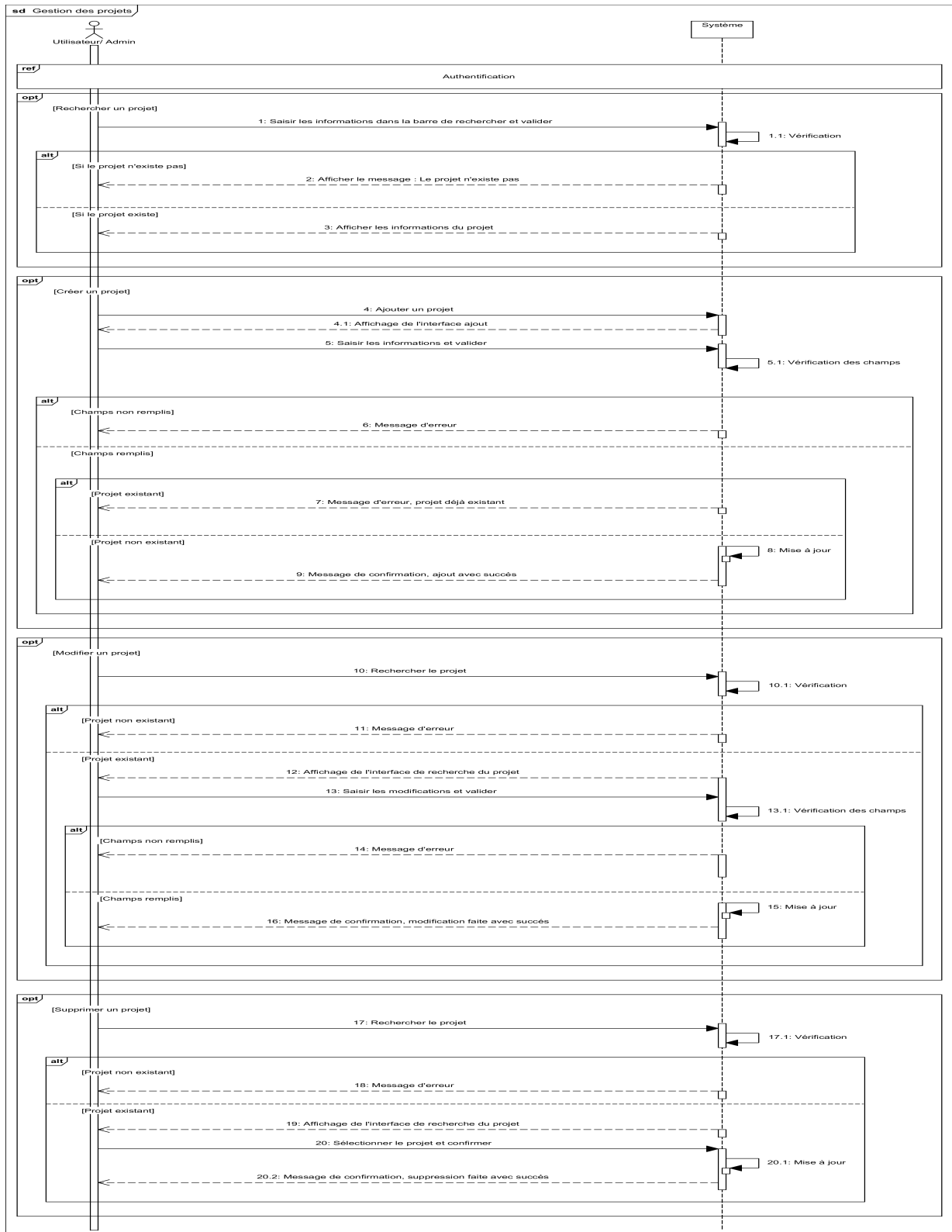


FIGURE 3.2 – Diagramme de séquence Gestion des projets.

3.2.3 Diagramme de séquence Gestion des Dashboard

Après s'être authentifié et avoir accès au projet correspondant, l'utilisateur accède à l'interface des Dashboards, où il peut effectuer les tâches suivantes : créer un Dashboard, modifier un Dashboard, supprimer un Dashboard ou rechercher un Dashboard.

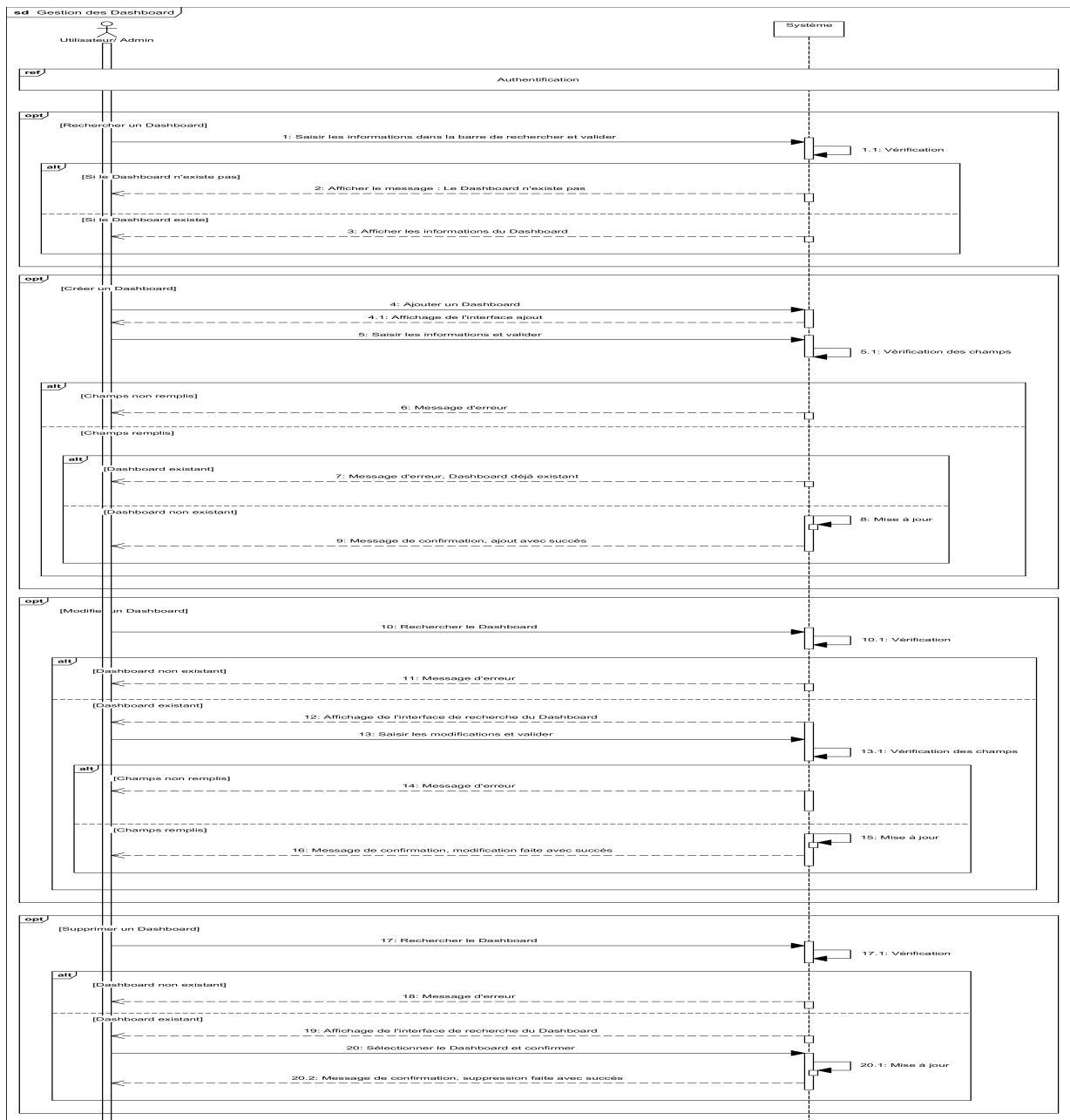


FIGURE 3.3 – Diagramme de séquence Gestion des Dashboard.

3.2.4 Diagramme de séquence Gestion des Graphes

Après s'être authentifié et avoir accès au Dashboard correspondant, l'utilisateur accède à l'interface des graphes, où il est peut effectuer les tâches suivantes : créer un graphe, modifier un graphe ou supprimer un graphe.

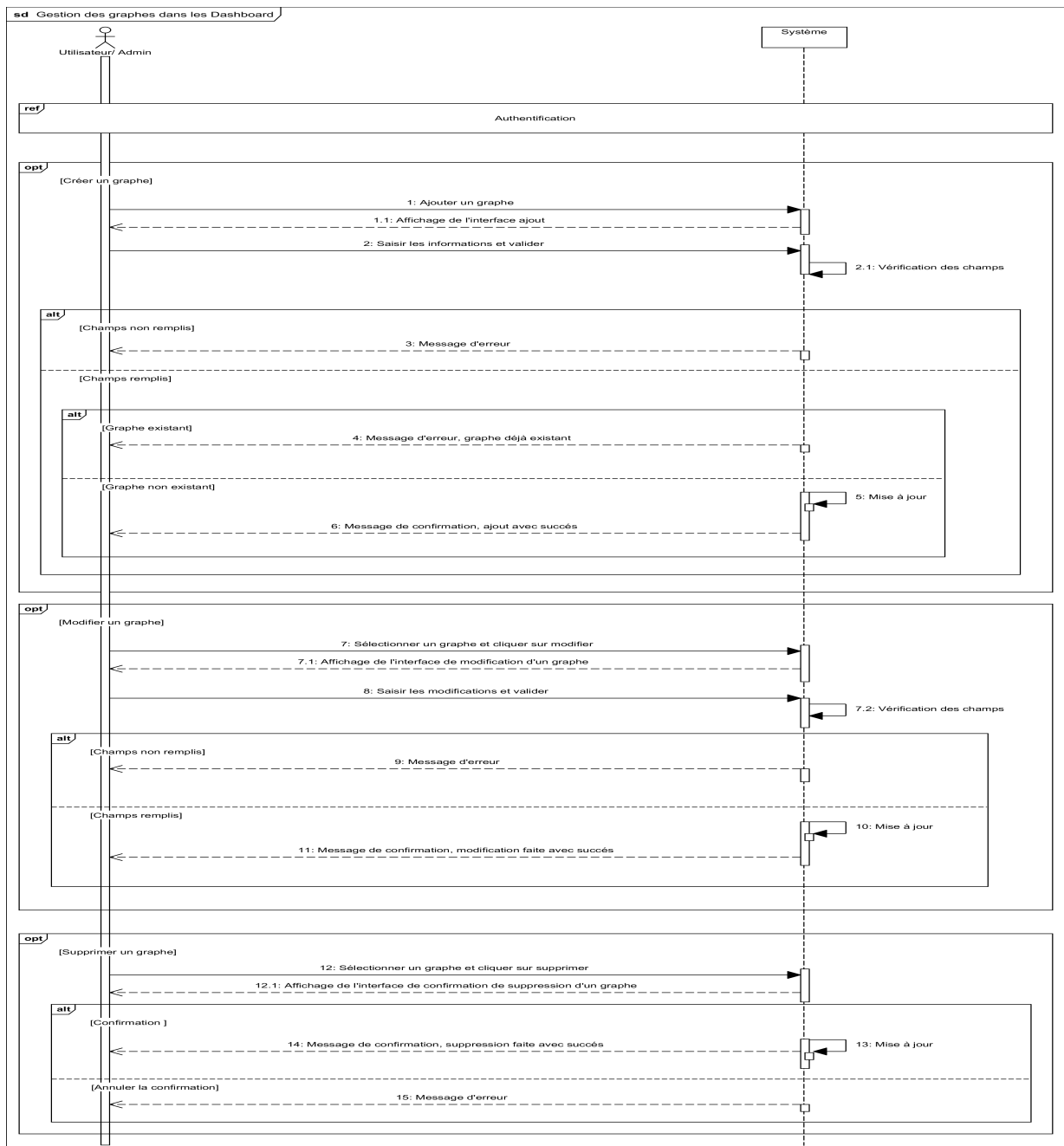


FIGURE 3.4 – Diagramme de séquence Gestion des graphes.

3.2.5 Diagramme de séquence Gestion des Comptes

Après s'être authentifié, l'administrateur accède à l'interface de la gestion des comptes, où il est peut effectuer les tâches suivantes : créer un compte, modifier un compte, supprimer un compte ou rechercher un compte.

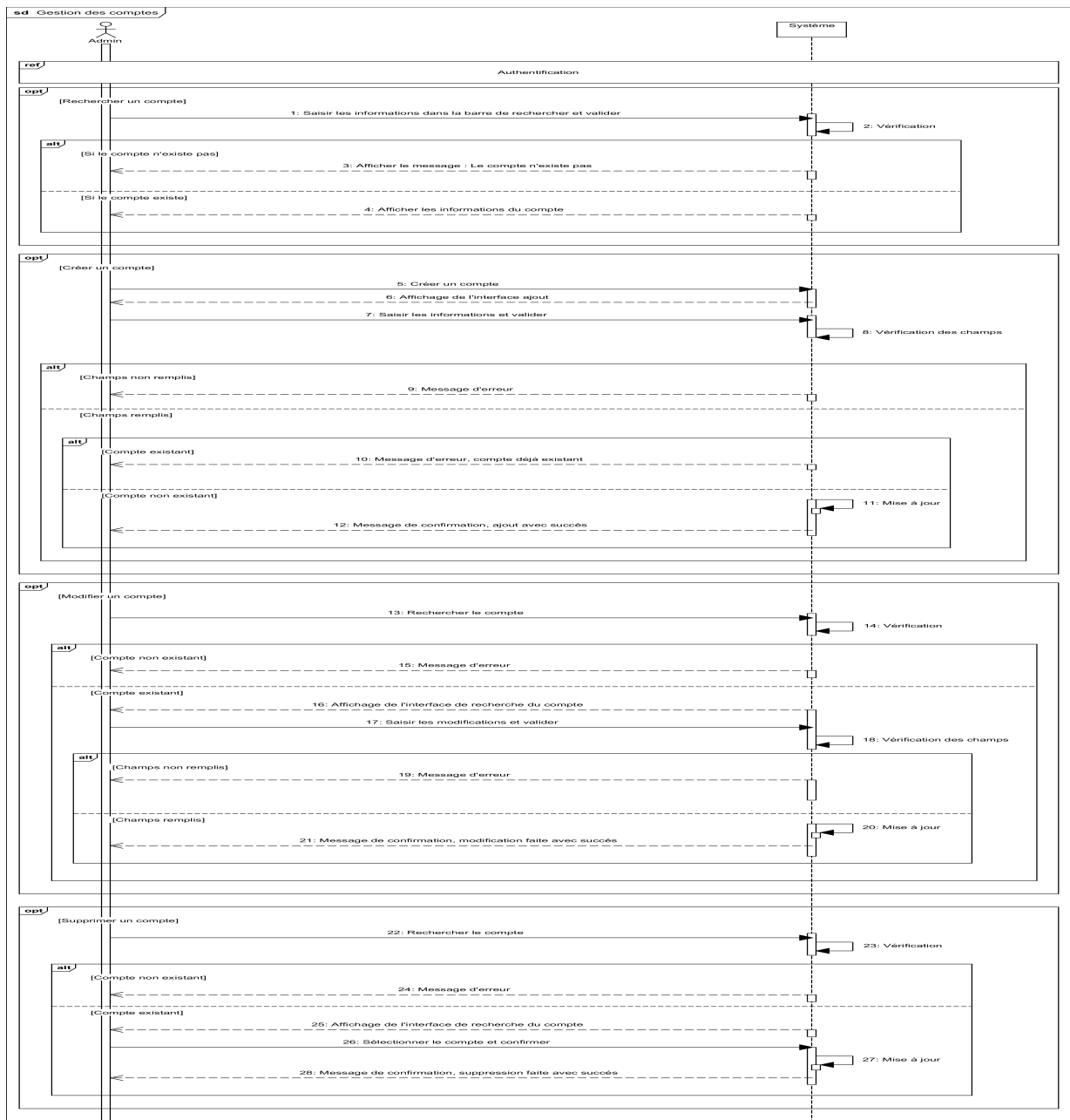


FIGURE 3.5 – Diagramme de séquence Gestion des comptes.

3.2.6 Diagramme de séquence Gestion visiteurs

Sur l'interface d'accueil, le visiteur peut visualiser la démonstrative de notre application et créer un compte s'il le souhaite.

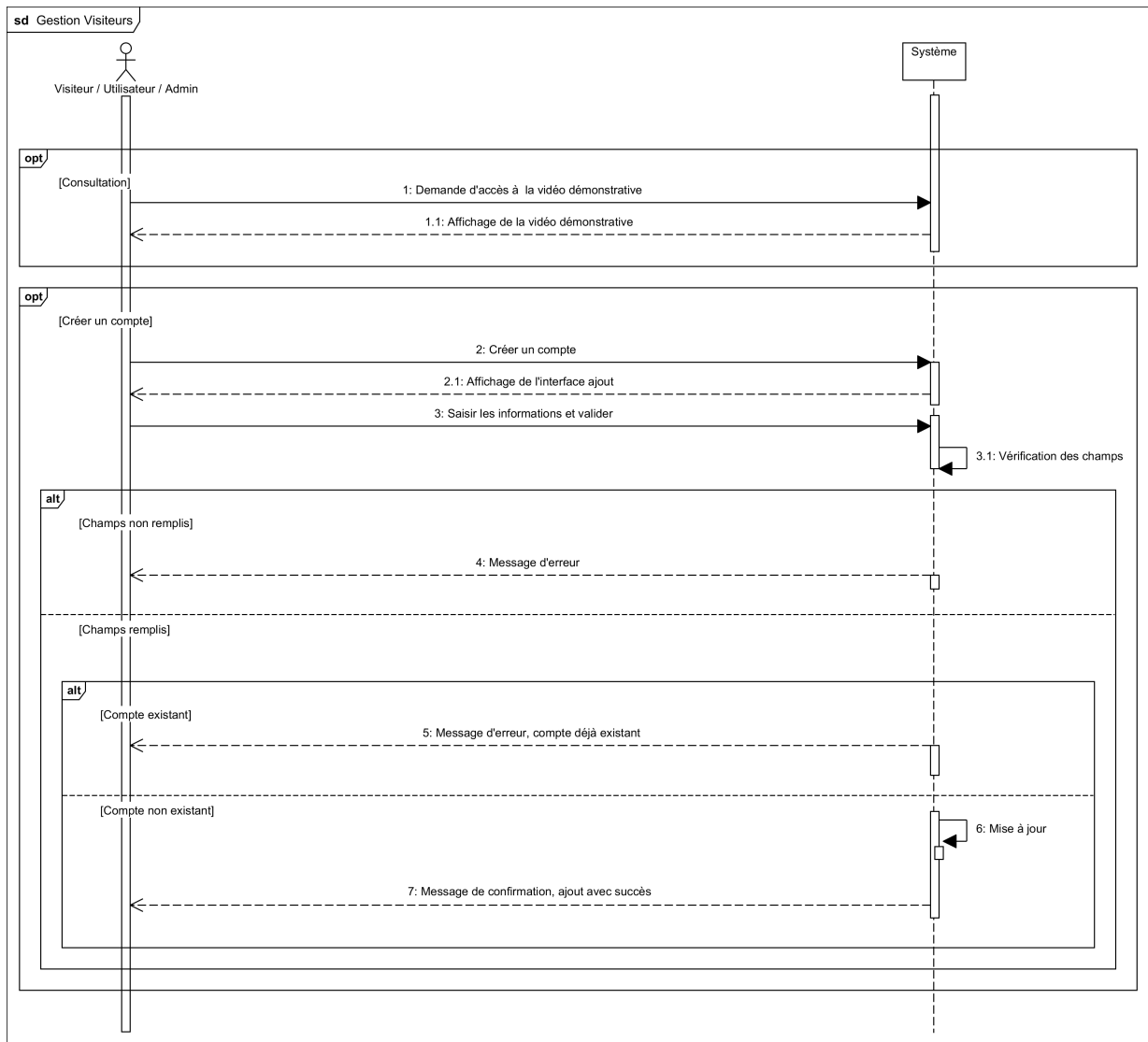


FIGURE 3.6 – Diagramme de séquence Gestion visiteurs.

3.3 Diagramme de classes

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objet de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet [15].

Ce type de diagramme représente les classes intervenant dans le système, chaque classe se décrit par des données dites attributs et de traitements dont elle est responsable vis-à-vis d'elle-même et dans d'autres classes, ainsi que les associations entre elles. La Figure 4.7 suivante montre le diagramme de classe final validé de notre système :

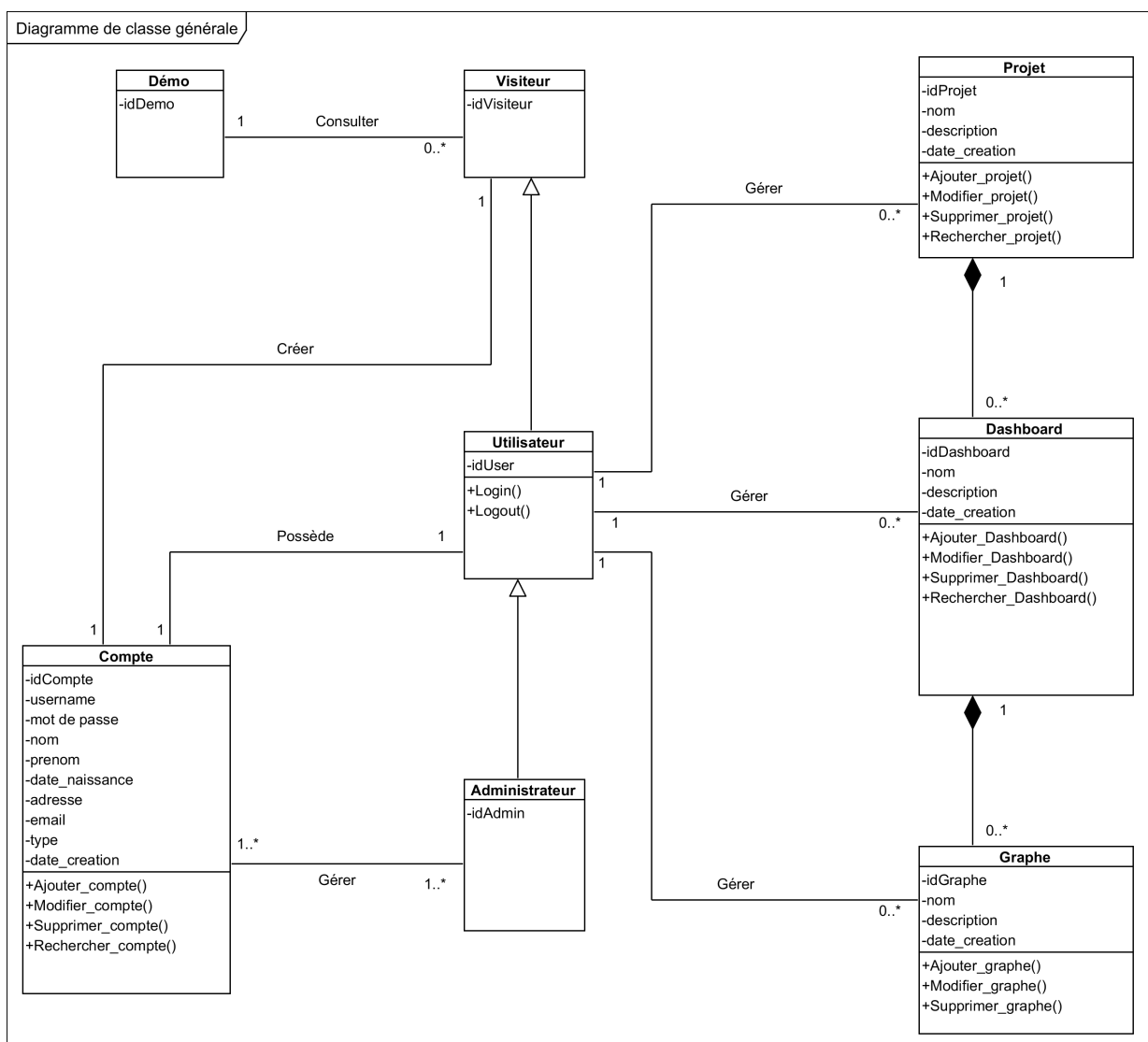


FIGURE 3.7 – Diagramme de classes.

3.4 Modèle relationnel

Le modèle relationnel est basé sur une organisation des données sous forme de tables. La manipulation des données se fait selon le concept mathématique de relation de la théorie des ensembles, c'est-à-dire l'algèbre relationnelle. Elle est constituée d'un ensemble d'opérations formelles sur les relations. Les opérations relationnelles permettent de créer une nouvelle relation (table) à partir d'opérations élémentaires sur d'autres tables (par exemple l'union, l'intersection, ou encore la différence) [16].

Afin de transformer le diagramme de classe en modèle relationnel, il est nécessaire d'appliquer les règles suivantes :

1. **Première règle**

Présence de la cardinalité (1..*) ou (0..*) d'un côté de l'association

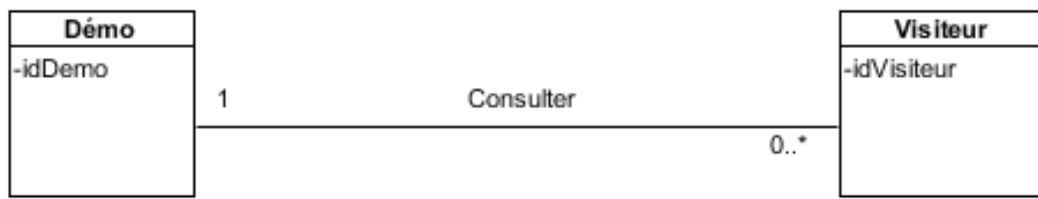


FIGURE 3.8 – Relation un à plusieurs.

- Chaque classe se transforme en une table.
- Chaque attribut de classe se transforme en un champ de table.
- L'identifiant de la classe qui est associé à la cardinalité (1) (ex : Démon) devient la clé étrangère de l'autre classe (ex : Visiteur).

2. Deuxième règle

Présence de la cardinalité (*..*) des deux côtés de l'association

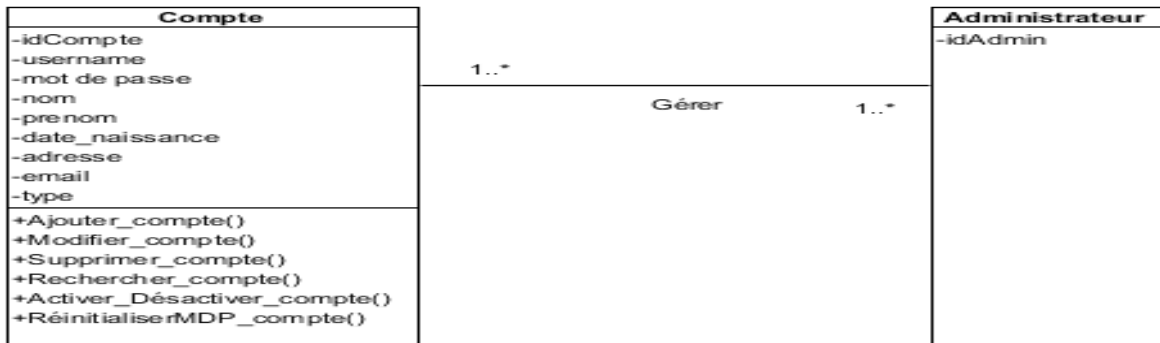


FIGURE 3.9 – Relation plusieurs à plusieurs.

- Chaque classe se transforme en une table.
- Chaque attribut de classe se transforme en un champ de table.
- L'association se transforme en une table. Cette table a comme champs l'identifiant de chacune des deux classes, plus d'éventuels autres attributs.

3. Troisième règle

Présence de la cardinalité (1..1) des deux côtés de l'association

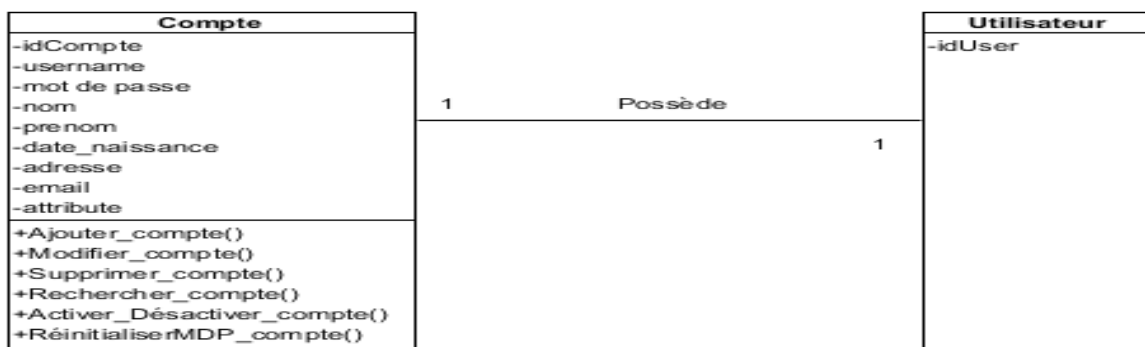


FIGURE 3.10 – Relation un à un.

- Chaque classe se transforme en une table.
- Chaque attribut de classe se transforme en un champ de table.
- L'identifiant de la classe qu'on souhaite, devient la clé étrangère de l'autre classe selon le besoin.

4. Quatrième règle

Présence d'une généralisation



FIGURE 3.11 – Relation généralisation.

- **Méthode 1 (push up) :**
 - Créer une table avec tous les attributs des classes.
 - Ajouter un attribut pour distinguer les types des objets.
- **Méthode 2 (push down) :**
 - Créer une table pour chaque sous type, chaque table se compose des attributs génériques et d'attributs spécifiques.
- **Méthode 3 :**
 - Créer une table par classe et des associations

Dans notre cas, toutes les règles ont été utilisées, car le diagramme de classes.

En suivant les règles de passage du diagramme de classes au modèle relationnel cités précédemment, on obtient le modèle relationnel suivant :

- **Démo** (idDemo)
- **Visiteur** (idVisiteur, #idDemo, idUser)
- **Utilisateur** (idUser, idAdmin)
- **Administrateur** (idAdmin)
- **Compte** (idCompte, username, mot de passe, nom, prenom, date_naissance, adresse, email, type, date_creation, #idUser, #idVisiteur)
- **Gérer** (#idAdmin , #idCompte)
- **Projet** (idProjet, nom, description, date_creation, #idUser)

- **Dashboard** (idDashboard, nom, description, date_creation, #idUser, #idProjet)
- **Graphe** (idGraphe, nom, description, date_creation, #idUser, #idDashboard)

3.5 Dictionnaire de données

Pendant la phase de conception, les données recueillies et spécifiées sont inscrites dans un dictionnaire de données qui est un document qui permet de recenser, de classer et de trier toutes les informations (les données) collectées lors des entretiens ou de l'étude des documents. Il constitue la référence de toutes les études effectuées par la suite [17].

Le Tableau 4.1 suivant illustre le dictionnaire de données de notre application :

Table	Attribut	Description de l'attribut	Type
Compte	idCompte	Identificateur	int
	username	Nom d'utilisateur	varchar
	mot de passe	Mot de passe d'utilisateur	varchar
	nom	Nom d'utilisateur	varchar
	prenom	Prénom d'utilisateur	varchar
	date de naissance	Date de naissance d'utilisateur	varchar
	adresse	Adresse d'utilisateur	varchar
	email	Email d'utilisateur	varchar
	type (Admin,User)	Type d'utilisateur	varchar
Utilisateur	idUser	Identificateur de l'utilisateur	int
Administrateur	idAdmin	Identificateur de l'administrateur	int
Visiteur	idVisiteur	Identificateur du visiteur	int
Démo	idDemo	Identificateur de la démo	int
Projet	idProjet	Identificateur du projet	int
	nom	Nom du projet	varchar
	description	Description du projet	varchar
	date creation	Date du création du projet	varchar

Dashboard	idDashboard	Identificateur du dashboard	int
	nom	Nom du dashboard	varchar
	description	Description du dashboard	varchar
	background	Couleur de l'arrière plan	varchar
	date creation	Date du création du dashboard	varchar
Graphe	idGraphe	Identificateur du graphe	int
	nom	Nom du graphe le dashboard	varchar
	description	Description du graphe le dashboard	varchar
	date creation	Date de création du graphe	varchar

TABLE 3.1 – Dictionnaire de données.

3.6 Conclusion

Ce chapitre a été consacré à la conception de l'application web de la data visualisation. En premier lieu nous avons détaillé l'aspect dynamique du système via des diagrammes de séquences système puis l'aspect statique via un diagramme de classe qui nous a permis d'aboutir au model relationnel en suivant les règles de passage d'un diagramme de classe au modèle relationnel et ce afin d'extraire les tables à implémenter dans la base de données.

A la fin de ce chapitre, nous avons jugé très utile de dresser le dictionnaire de données qui nous facilitera par la suite la création des tables, la codification des attributs avec les détails nécessaires de ces derniers.

Désormais, la phase de conception est terminée, nous pouvons s'étaler alors sur l'activité finale de notre projet qui est la phase de réalisation.

Réalisation

4.1 Introduction

Après avoir abordé la partie analyse et conception appropriée à notre projet, nous allons entamer la dernière phase de notre projet qui est la réalisation de notre application. Dans ce chapitre, nous présenterons l'environnement et les outils de développement . Nous passerons par la suite à la présentation des différentes interfaces de notre application ainsi que les étapes que nous avons suivi pour son déploiement.

4.2 Application web

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Une application web manipule des fichiers qui sont écrits en langage HTML (les pages web). Ces pages sont stockées sur un serveur et sont exécutées par le navigateur web du client.

Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur favori. La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs.

Les technologies utilisées pour développer les applications web sont les mêmes que celles employées dans la création des sites internet [18].

4.3 Outils de développement

4.3.1 Visual studio code

Visual studio code est un éditeur de code léger mais puissant qui s'exécute sur un bureau et est disponible sur Windows, macOS et linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et possède un écosystème riche d'extensions pour d'autres langages (tels que C++, C#, Java, Python, PHP, Go) et les environnements d'exécution (tels que .NET et Unity) [19].

4.3.2 Visual Paradigm

Visual Paradigm for UML est comme son nom le laisse supposer, un logiciel permettant aux programmeurs de mettre en place des diagrammes UML. Disposant d'un outil créant des rapports personnalisables aux formats PDF, Word ou HTML afin de les partager et les publier sur Internet, cette application est compatible avec de nombreuses applications, standards et environnements. Ainsi, il est possible de générer notamment des diagrammes de séquences ou de cas d'utilisation et ainsi produire du code source dans de nombreux langages comme le Java ou encore le C++, ou bien faire l'inverse, générer des diagrammes à partir de code déjà existant [20].

4.3.3 Wampserver

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données [21].

4.4 Langages de programmation

Une grande variété de langages de programmation web existe sur le net. Le choix peut se porter également sur des frameworks de développements web qui regroupent des fonctionnalités très intéressantes. Dans le cadre de notre projet, nous avons fait le choix des langages et des outils ci-dessous :

4.4.1 Bootstrap

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option [22].

4.4.2 MySQL (My Structured Query Language)

MySQL est un système de gestion de bases de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il permet de traiter et de maintenir de grosses bases de données avec une grande fiabilité. [23]

4.4.3 Git

Git est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. Git est un projet open source avancé, qui est activement maintenu. À l'origine, il a été développé en 2005 par Linus Torvalds, le créateur bien connu du noyau du système d'exploitation Linux. De plus en plus de projets de développement reposent sur Git pour le contrôle de version, y compris des projets commerciaux et en open source. Les serveurs de Git sont : Github, GitLab , Bitbucket [24].

4.4.4 JavaScript

JavaScript est un langage de script ou de programmation qui permet d'implémenter des programmes complexes sur les pages web, ces scripts peuvent être écrits directement sur une page web HTML et exécutés automatiquement quand la page se charge [25].

4.4.5 HTML (Hypertext Markup Language)

L'HTML est un langage informatique largement utilisé dans la création des pages web. Son acronyme signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.

Ce n'est pas à proprement parlé un langage de programmation, mais plutôt un langage qui permet de mettre en forme du contenu. Les balises permettent de mettre en forme

le texte et de placer des éléments interactifs, tel des liens, des images ou bien encore des animations [22].

4.4.6 CSS

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML [26].

4.4.7 Angular

C'est un framework d'application Web gratuit et open source basé sur TypeScript dirigé par l'équipe Angular de Google et par une communauté de particuliers et les sociétés [27].

4.4.8 TypeScript

TypeScript est un langage gratuit et open-source développé et maintenu par Microsoft depuis octobre 2012. TypeScript est une surcouche d'ECMAScript permettant l'ajout optionnel de typage statique. Il n'existe actuellement pas de réel runtime TypeScript. Il faut utiliser un compilateur pour transpiler le code TypeScript en code ECMAScript valide que l'on peut ensuite exécuter sur le runtime JavaScript de notre choix : Browser, NodeJS etc... TypeScript est un parfait compromis entre la flexibilité d'un langage dynamiquement typé et la rigueur d'un langage statiquement typé sans tomber dans la lourdeur syntaxique associée et ce grâce à d'intéressants concepts tels que le "duck typing" ou l'inférence de type [28].

4.4.9 Ngx-Charts

Ngx-Charts est un framework de graphique open source et déclaratif pour angular2+. Il est entretenu par Swimlane. Il utilise Angular pour rendre et animer les éléments SVG avec toutes ses qualités de liaison et de vitesse et utilise d3 pour les excellentes fonctions mathématiques, échelles, générateurs d'axes et de formes, etc [29].

4.4.10 Swimlane

Swimlane est un leader en matière d'orchestration, d'automatisation et de réponse de sécurité (SOAR). En automatisant les processus manuels et les workflows opérationnels chronophages et en fournissant des analyses puissantes et consolidées, des tableaux de bord en temps réel, Swimlane maximise les capacités de réponse aux incidents des opérations de sécurité surchargées et en sous-effectif [30].

4.4.11 NodeJs

Node.js est un environnement open-source permettant de développer des applications multiplateformes. Basé sur le V8 de Google, le moteur d'exécution JavaScript utilisé dans Chrome, Node.js traite donc les données en JavaScript. Node.js permet d'utiliser JavaScript côté serveur, en dehors du navigateur avant de pouvoir exécuter le code JavaScript sur le serveur, le code a besoin d'être interprété. C'est précisément ce que fait Node.js en utilisant le moteur V8 de Google, celui qui est utilisé par le navigateur Chrome. D'autre part, Node.js intègre un nombre important de modules utiles, donc, pas besoin d'écrire tout le code à partir de zéro. Ainsi, Node.js comprend deux éléments distincts : un contexte d'exécution et une bibliothèque [31].

4.4.12 Node Package Manager (NPM)

NPM (Node Package Manager) comme son nom l'indique est le "package manager" officiel de l'univers JavaScript (frontend /backend). Il est installé automatiquement lors de l'installation de NodeJS [32].

4.4.13 JSON

JSON (JavaScript Object notation) est un format léger qui permet de décrire des objets en JavaScript, il est principalement utilisé pour faire des échanges entre JS et un serveur web. JSON se base sur deux structures :

- Une collection de couples nom/valeur : {nom : valeur ;}
- Un tableau qui est une collection de valeurs ordonnées [valeur, valeur]

JSON est un format de données. Autrement dit, c'est une façon de stocker des informations, un peu comme une base de données. Il est maintenant lié à JavaScript qui inclut un objet JSON, et de nombreux développeurs l'incorporent quasiment comme un sous-ensemble du langage [33].

4.4.14 Express

Express.js , ou simplement Express , est un framework d'application Web back-end pour Node.js , publié en tant que logiciel gratuit et open source sous la licence MIT . Il est conçu pour créer des applications Web et des API, c'est le framework de serveur standard de facto pour Node.js [34].

4.5 Notre application de gestion de la DataViz

L'interface graphique est le moyen d'échange entre l'humain et la machine, il est important qu'elle soit bien conçue, ergonomique et intuitive afin que l'utilisateur la comprenne facilement et lui rend la tâche facile. Dans ce qui suit une présentation des captures d'écrans des interfaces que nous avons jugés les plus importantes.

4.5.1 Interface d'authentification

Cette page est un élément primordiale de l'application, elle représente la phase d'identification pour accéder à ses fonctionnalités. Dès la connexion, la page d'authentification s'affiche, et elle est composée de la zone «login» et de la zone «mot de passe», l'utilisateur accède à la page qui lui est dédiée selon son rôle :

- Administrateur
- Utilisateur

La figure 4.1 suivante représente l'interface d'authentification de la plateforme où l'utilisateur s'authentifie pour accéder à l'application.

FIGURE 4.1 – Interface d’authentification.

4.5.2 Interface Inscription

L’utilisateur de la plateforme a la possibilité de s’inscrire s’il n’est pas encore enregistré dans la base de donnée, et ceci en cliquant sur le lien « inscription » présent dans l’interface d’authentification, qui va le rediriger vers une autre page (Figure 4.2), où il saisit ses coordonnées puis valide son envoi. Après vérification , le système affiche un message de confirmation.

FIGURE 4.2 – Interface d’inscription.

4.5.3 Gestion des Utilisateurs

4.5.3.1 Interface " liste des utilisateurs "

La figure 4.3 représente l'interface affichée à l'administrateur après son authentification, qui va lui permettre de créer, modifier, supprimer ou rechercher un utilisateur.

saisir un firstname								+
id	firstname	lastname	adresse	date de naissance	type User	email	username	
18	bbbbbbbbbb	aaaaa	aaaaaaaaa	02/02/2016	user	billalkebbi@gmail	billaladminaa	
8	adamo	adamo	bejaia	01/10/1999	admin	adem@gmail.com	ademadem	
26	brahh	brahii	bejaia	02/01/1996	user	brah@gmail.com	brabraadmin	
25	brahim	brahim	bejaia	01/01/2010	user	brahim@gmail.com	brahimadmin	
23	chafaa	chafaa	bejaia	01/01/2007	user	chafaa@gmail.com	chafaauser	
27	chaaff	CHAFFF	bejaia	01/02/2006	user	chaf@gmail.com	chafadmin	
28	djamal	djamal	bejaia	01/01/2007	admin	br@gmail.com	djamaladmin	
7	billal	kebbi	seddou	30/12/1997	admin	billal@gmail.com	billaladmin	

FIGURE 4.3 – Interface " liste des utilisateurs ".

4.5.3.2 Interface " Ajouter un utilisateur "

La figure 4.4 représente l'interface affichée à l'administrateur pour lui permettre d'ajouter un utilisateur.

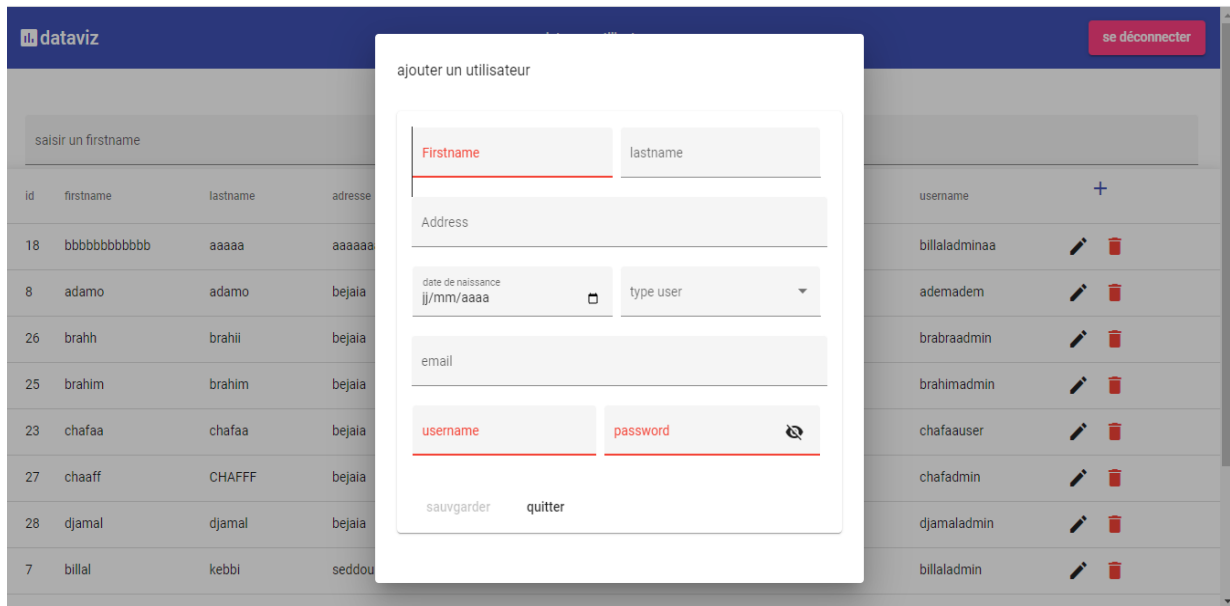


FIGURE 4.4 – Interface " Ajouter un utilisateur " .

4.5.3.3 Interface " Modifier un utilisateur "

La figure 4.5 représente l'interface affichée à l'administrateur pour lui permettre de modifier un utilisateur.

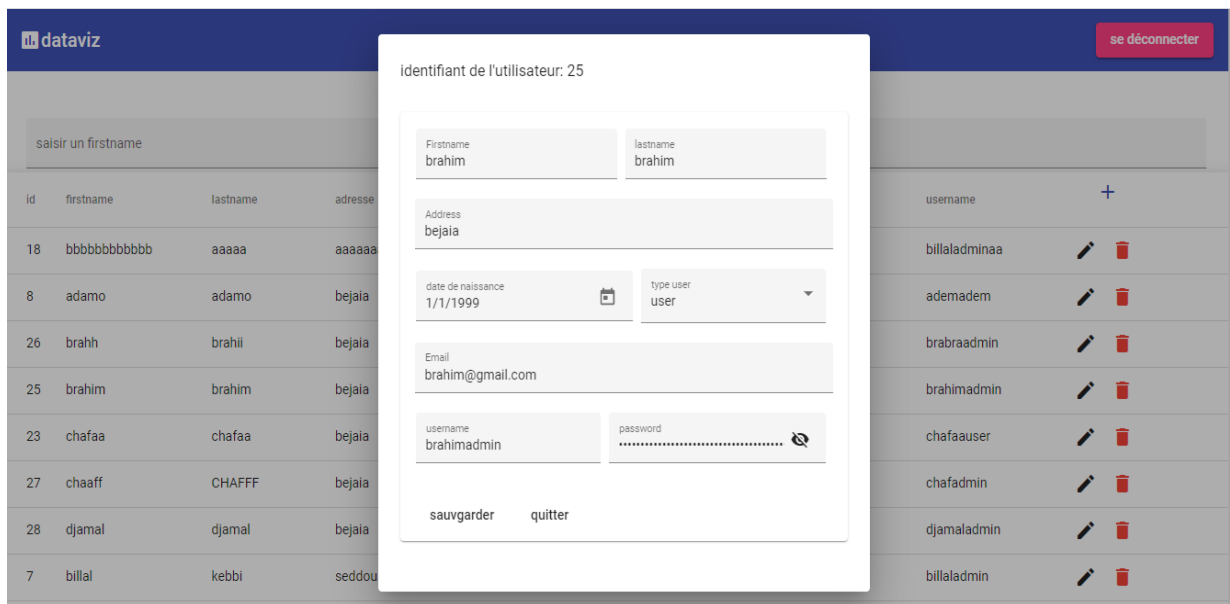


FIGURE 4.5 – Interface " Modifier un utilisateur " .

4.5.3.4 Interface " Supprimer un utilisateur "

La figure 4.6 représente l'interface affichée à l'administrateur pour lui permettre de supprimer un utilisateur.

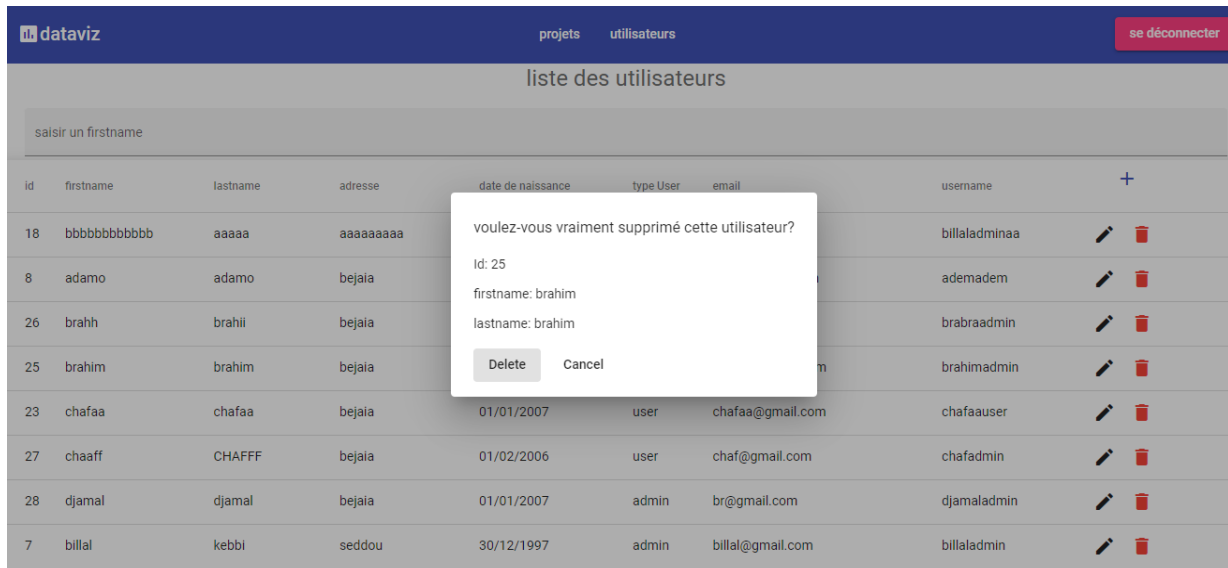


FIGURE 4.6 – Interface " Supprimer un utilisateur ".

4.5.4 Gestion des Projets

4.5.4.1 Interface " liste des projets "

La figure 4.7 représente l'interface affichée à l'utilisateur après son authentification et qui va lui permettre de créer, modifier, supprimer ou rechercher un projet.

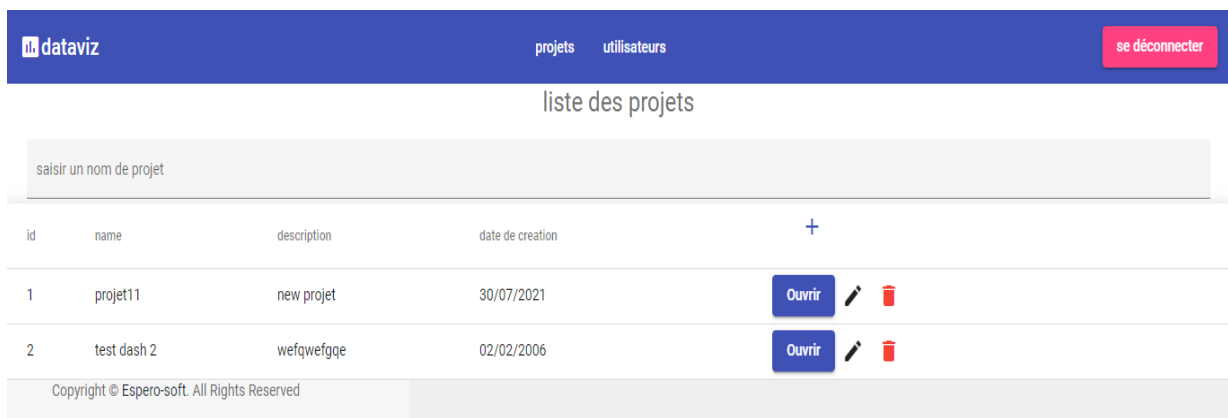


FIGURE 4.7 – Interface " liste des projets ".

4.5.4.2 Interface " Ajouter un projet "

La figure 4.8 représente l'interface affichée à l'utilisateur qui permet à ce dernier de créer un projet en introduisant le nom et la description du projet.

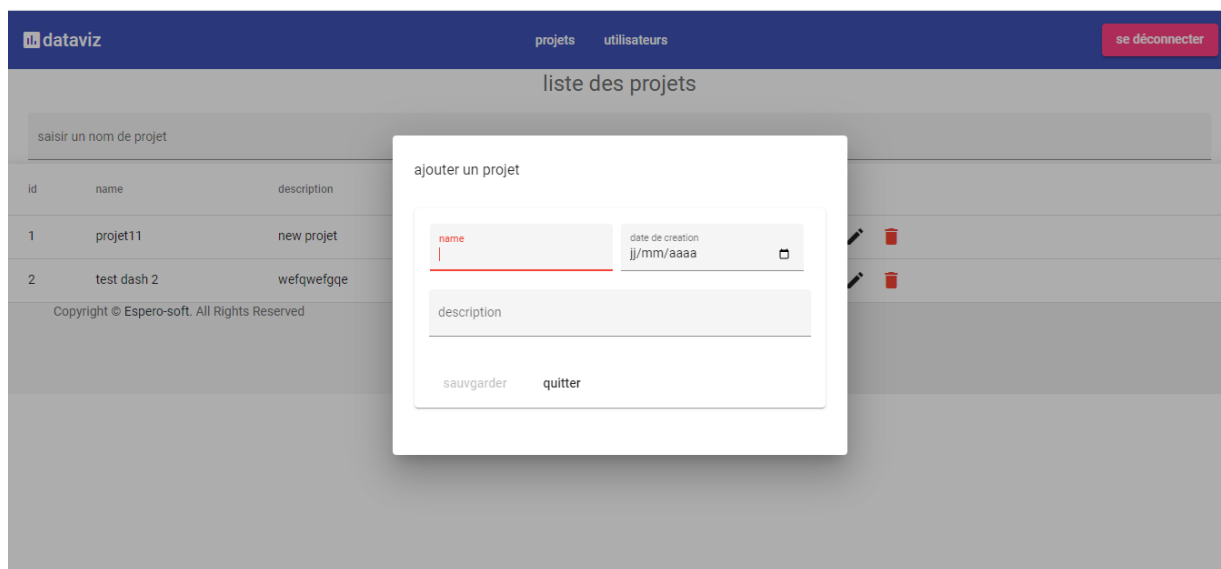


FIGURE 4.8 – Interface " Ajouter un projet " .

4.5.4.3 Interface " Modifier un projet "

Afin de modifier un projet, l'utilisateur accède à l'interface représentée dans la figure 4.9 .

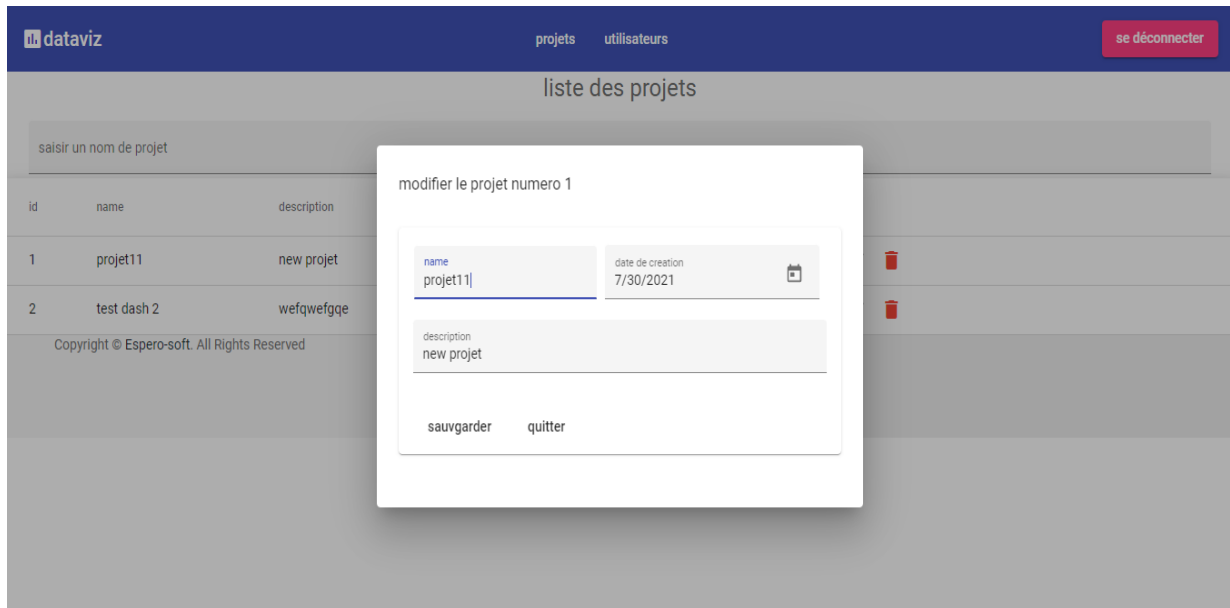


FIGURE 4.9 – Interface " Modifier un projet " .

4.5.4.4 Interface " Supprimer un projet "

Afin de supprimer un projet, l'utilisateur choisi le projet et clique sur le bouton supprimer qui va lui afficher l'interface représentée dans la figure 4.10 .

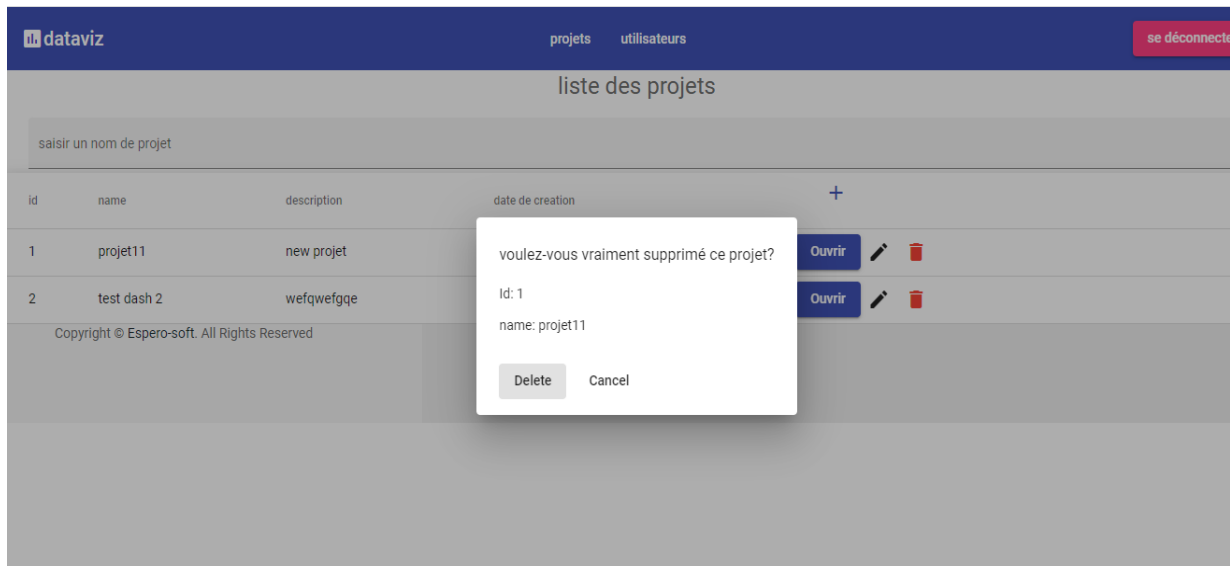


FIGURE 4.10 – Interface " Supprimer un projet " .

4.5.5 Gestion des Dashboards

4.5.5.1 Interface " liste des Dashboards "

La figure 4.11 représente l'interface sur laquelle il sera possible de créer, modifier, supprimer ou rechercher un Dashboard.

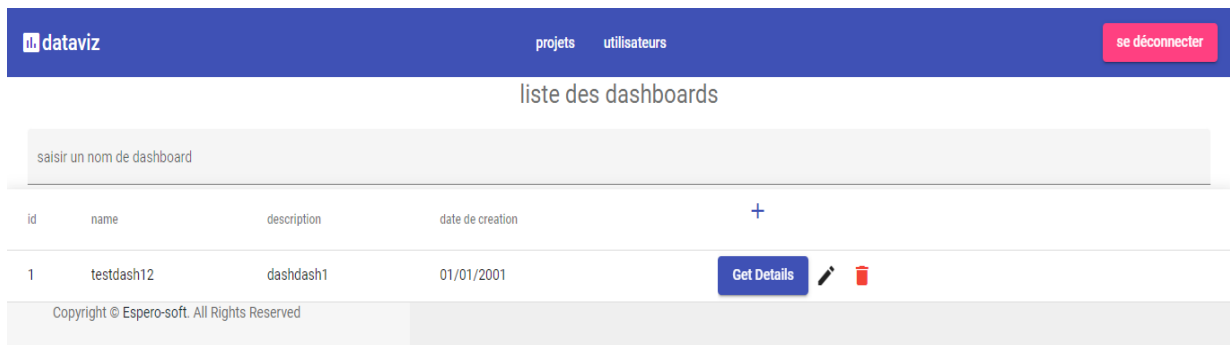


FIGURE 4.11 – Interface " liste des Dashboards ".

4.5.5.2 Interface " Ajouter un Dashboard "

La figure 4.12 représente l'interface affichée à l'utilisateur pour lui permettre de créer un Dashboard.

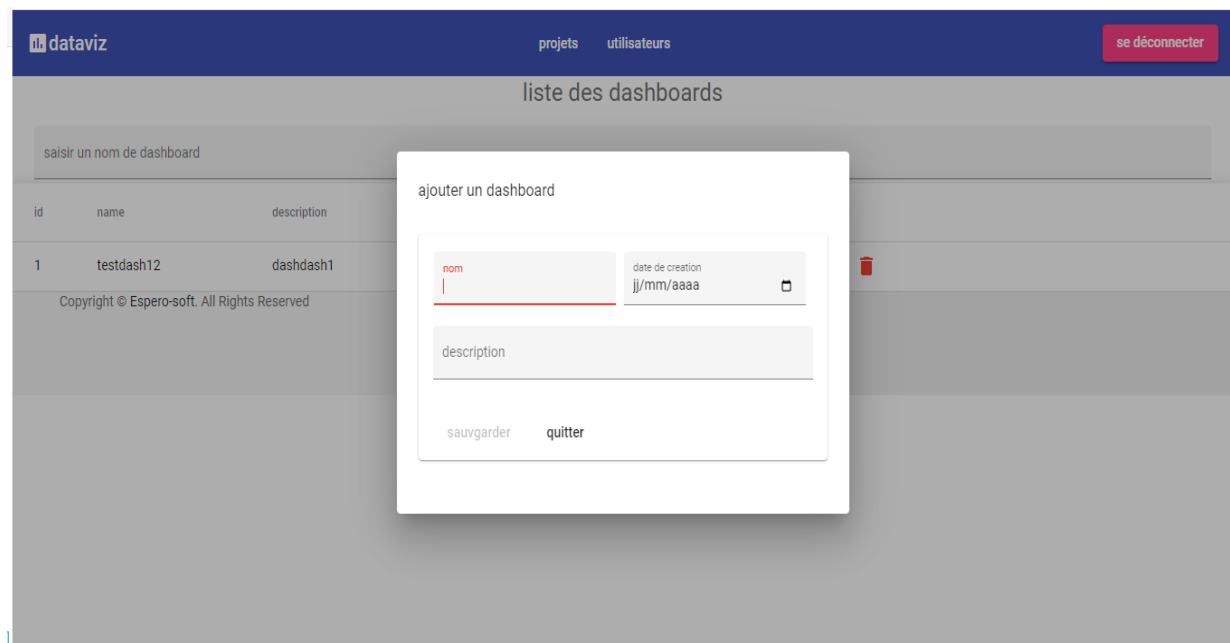


FIGURE 4.12 – Interface " Ajouter un Dashboard ".

4.5.5.3 Interface " Modifier un Dashboard "

La figure 4.13 représente l'interface affichée à l'utilisateur pour lui permettre de modifier un Dashboard.

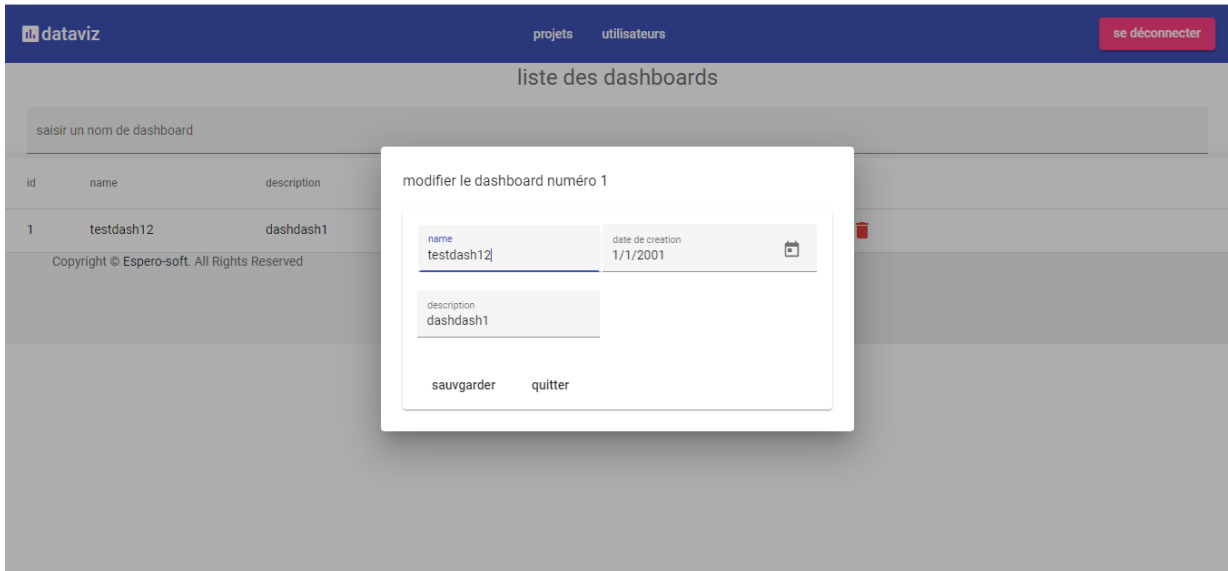


FIGURE 4.13 – Interface " Modifier un Dashboard ".

4.5.5.4 Interface " Supprimer un Dashboard "

La figure 4.14 représente l'interface affichée à l'utilisateur pour lui permettre de supprimer un Dashboard.

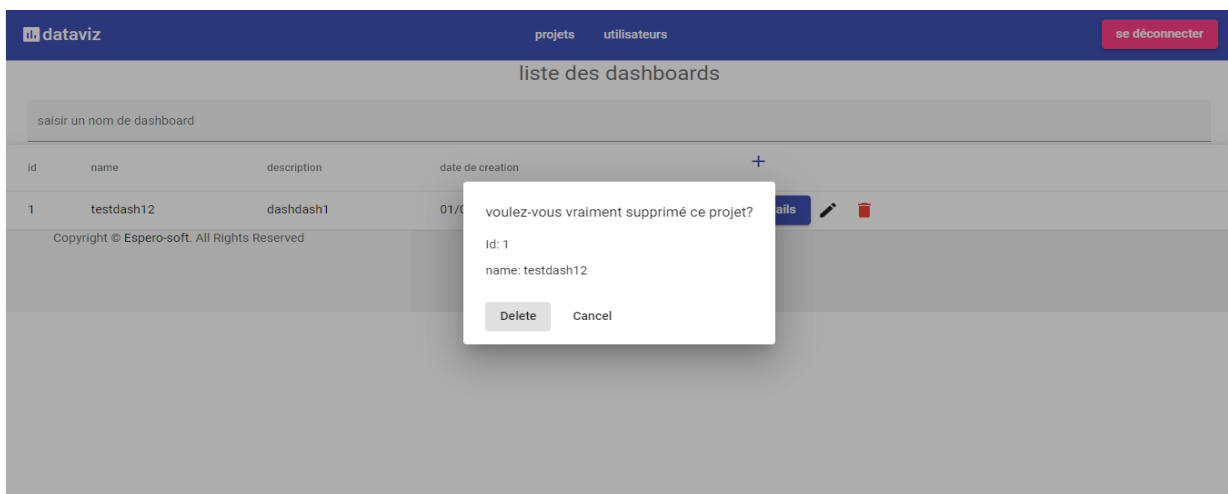


FIGURE 4.14 – Interface " Supprimer un Dashboard ".

4.5.6 Gestion des Graphes

4.5.6.1 Interface " Ajouter un graphe dans un Dashboard "

La figure 4.15 représente l'interface affichée à l'utilisateur , lui permettront de créer un graphe dans un Dashboard.

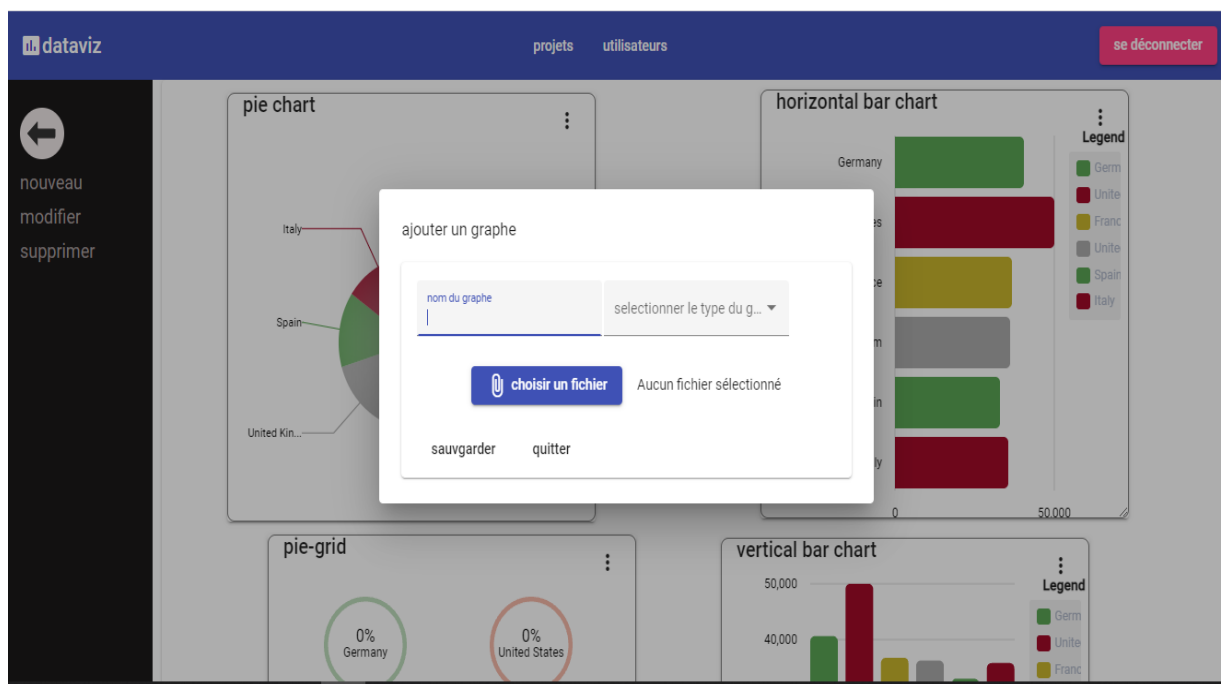


FIGURE 4.15 – Interface " Ajouter un graphe dans un Dashboard ".

4.5.6.2 Interface " Visualisation des graphes dans un Dashboard "

La figure 4.16 représente l'interface affichée à l'utilisateur pour permettre d'afficher les graphes dans le Dashboard.



FIGURE 4.16 – Interface " Visualisation des graphes dans un Dashboard ".

4.6 Conclusion

Dans ce chapitre réservé à la partie développement d'une application web de la data visualisation de l'entreprise Tech-Instinct, nous avons commencé par la présentation des langages et des outils de développement web que nous avons utilisé. Nous avons présenté, également, Angular et Node.js, très utilisé actuellement par les grandes entreprises de développement web dans le monde. Par ailleurs, nous avons présenté quelques interfaces de notre application pour montrer l'implémentation de quelques cas d'utilisation cités dans le chapitre 3.

Conclusion générale

Ce projet de fin de cycle avait comme objectif la conception et l'implémentation d'une plateforme de data visualisation pour l'entreprise Tech'Instinct.

Nous avons établie un état de l'existant pour nous baser sur une bonne analyse grâce auquel nous avons pu capturer tous les besoins des futurs utilisateurs de la plateforme, ce qui nous a permis de fixer la structure globale de l'application à partir la phase de conception.

L'accomplissement de cette mission, nous a aidé à approfondir et élargir nos connaissances acquises durant notre cursus universitaire, à pratiquer et à utiliser des nouvelles technologies de développement, et ceci en suivant des méthodes de modélisation et de conception fondamentales.

Nous sommes assez certains, que cet outil de data visualisation, sera bénéfique et apportera du dynamisme et va permettre d'améliorer et d'optimiser du temps au prochains utilisateurs afin d'analyser leurs données plus facilement.

Enfin, nous espérons que ce modeste travail pourra servir de référence à tous ceux qui voudront s'engager dans un travail similaire et homologue.

Perspectives

Pour la conception de notre application, nous avons utilisé le processus de développement UP et le langage UML pour la réalisation des différents diagrammes couvrant toutes les phases du processus de développement, offrant ainsi une base solide sur laquelle nous nous sommes appuyés afin de passer à la partie réalisation du projet.

Pour la mise en œuvre de l'application, nous avons eu recours à la programmation avec Angular comme front-end et Node.js comme back-end et Ngx-Charts comme système de gestion de graphe et le SGBD MySQL. Notre réalisation n'est qu'une première version, d'actualité et ne s'arrête pas à ce niveau.

En effet plusieurs perspectives s'offrent à ce projet tel que :

- Ajouter un système de génération de l'iframe du Dashboard
- Ajouter un système d'exportation du Dashboard sous format PDF . . .

Cette application pourrait être améliorée, élargie et enrichie de plus de fonctionnalités afin de répondre aux besoins de tous ses utilisateurs et mettre fin aux problèmes de données.

Enfin ce stage, ainsi que le travail fournit, nous a permis d'avoir un aperçu sur le monde du travail et développer nos compétences en tant que développeur mais aussi notre capacité à travailler en équipe.

Bibliographie

- [1] Dataviz 30 cas concret pour comprendre la data visualization, 2013, Electronic Business Group / Elenbi et SAS.
- [2] Revue internationale d'éducation de Sèvres, 66 | septembre 2014
<https://journals.openedition.org/ries/3949>.
- [3] Data Visualisation : De l'extraction des données à leur représentation graphique, Nathan Yau, 2013, Eyrolles.
- [4] <https://junto.fr/blog/data-visualisation/>.
- [5] <https://www.cartelis.com/blog/outils-data-visualisation/>.
- [6] Revue Scientifique
Datavisualisation, Des données à la connaissance, I2D - Information, données et documents 2015/2.
- [7] Data Visualisation : A Handbook for Data Driven Design, Andy Kirk, 2016, Sage Publishing.
- [8] Manuel de DATA VISUALISATION , Jean-Marie Lagnel, 21 juin 2017, DUNOD.
- [9] Data Visualization Techniques From Basics to Big Data With SAS ® Visual Analytics.
- [10] The Functional Art : An introduction to information graphics and visualization, Alberto Cairo, 2012, New Riders.
- [11] Data Visualization : A Successful Design Process, Andy Kirk, 2012, Packt Publishing.
- [12] Design for Information, Isabel Meirelles, 2013, Rockport.
- [13] Infographics : The Power of Visual Storytelling, Jason Lankow, Josh Ritchie, Ross Crooks, 2012, Wiley.
- [14] <https://tech-instinct.com/>.
- [15] Pascal Roques, UML 2 : Modéliser une application web, Les Cahiers du Programmeur, Eyrolles ,4e édition, Paris, 2008.

- [16] Introduction au modèle relationnel, <https://web.maths.unsw.edu.au/~lafaye/CCM/relation/relintro.htm>.
- [17] Kass Kasongo Ndjoko, Conception d'un système d'information pour le contrôle des structures des écoles, https://www.memoireonline.com/03/15/8981/m_Conception-d-un-systeme-d-information-pour-le-contrle-des-structures-des-ecoles-11.html, 2012.
- [18] Application web, sur le site Idéematic, <https://www.ideematic.com/dictionnaire-digital/application-web/>,.
- [19] FILEHIPPO. VISUAL STUDIO CODE
<https://filehippo.com/fr/>.
- [20] Publié par La Rédaction, Visual Paradigm for UML : Un logiciel de conception UML très complet, <http://www.clubic.com/telecharger-fiche384046-visual-paradigm-for-uml.htm>.
- [21] 'Wamp server Guide | Wamp server'.
<https://www.wampserver.com/>.
- [22] JDN. PHP (Hypertext Preprocessor) : définition
<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203597-php-hypertext-preprocessor-definition/?fbclid=IwAR0Uh2RiI-QIPVUxs0oJ6JmxR1x-8bSgU-AdlRBS5sP7aoGUbPCToHzvAzY>.
- [23] CHANTAL GRIBAUTMONT (OPENCLASSROOM). ADMINISTRER VOS BASES DE DONNEES AVEC MySQL
openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql?fbclid=IwAR3xAT7h-icCIPXiH6dlfcvdXm1aGVxttt6m-CvX9olW_e3tyNIL2gpGfno.
- [24] Qu'est-ce que Git ?,
<https://www.atlassian.com/fr/git/tutorials/what-is-git>,.
- [25] Qu'es-ce que JavaScript ?
<https://en.wikipedia.org/wiki/JavaScript>.
- [26] FUTURA-SCIENCES. CSS
www.futura-sciences.com/tech/definitions/internet-css-4050/?fbclid=IwAR3xAT7h-icCIPXiH6dlfcvdXm1aGVxttt6m-CvX9olW_e3tyNIL2gpGfno.
- [27] Qu'es-ce que Angular ?,
<https://angular.io/docs>.
- [28] GitBook
<https://guide-angular.wishtack.io/typescript>.

- [29] Qu'es-ce que Ngx-Charts ?
<https://www.ngdevelop.tech/how-to-use-ngx-charts-in-angular/amp/>.
- [30] Qu'es-ce que Swimlane ?
https://en.wikipedia.org/wiki/Swim_lane.
- [31] 'Node.js Guide | Node.js'.
<https://nodejs.org>.
- [32] Younes JAAIDI. 'Le guide angular'.
<https://guideangular.wishtack.io/tools/npm:text=NPM%20permet%20entre%20autres%20%3A,fig%20hier%20de%20description%20de%20d%C3%A9pendances>.
- [33] [AIT HATRIT Fatima et all, Conception et réalisation d'un site web dynamique pour l'aéroport « ABANE RAMDANE » De Bejaia, Mémoire de licence, université ABDE-RAHMANE MIRA, 2014/2015.].
- [34] Qu'es-ce que Express ?
<https://expressjs.com/en/5x/api.html>.

Résumé

Les entreprises sont amenées à extraire des informations en analysant leurs données, ces tâches n'étant pas facile, à cause du volume des données et leurs formats de représentation. C'est pourquoi les entreprises se sont retournées vers la data visualisation, qui consiste à représenter les données sous format graphique pour faciliter l'analyse et l'extraction d'informations.

Ce travail a pour but l'obtention du diplôme master professionnel en génie logiciel. L'objectif est de concevoir, et implémenter une plateforme de data visualisation, qui permettra au utilisateurs principalement de représenter graphiquement leurs données qui étaient initialement dans des fichier CSV, pour former des tableaux de bord (Dashboard) comportant un ou plusieurs graphiques, les utilisateurs auront aussi la possibilité de gérer leurs projets, et les administrateurs en plus des tâches que peut réaliser un simple utilisateur ils peuvent gérer la liste des utilisateurs. Ce dernier a été réalisé en utilisant le langage de modélisation Unified Modeling Language (UML) qui permet de schématiser la solution ainsi que les langages de programmation Angular, Node.js, Ngx-Charts et MySQL comme Système de Gestion de base de données (SGBD).

Mots clés : *UML, Angular, Node.js, Ngx-Charts, MySQL.*

Abstract

Companies have to extract information by analyzing their data. These tasks are not easy because of the volume of data and its representation formats. This is why companies have turned to data visualization, which consists of representing data in graphic format to facilitate analysis and information extraction.

This work aims to obtain a professional Master Degree in software engineering. The objective is to design, and implement a data visualization platform, which will allow users mainly to graphically represent their data that was initially in CSV files, to form dashboard with one or more graphs, users will also have the ability to manage their projects, and administrators in addition to the tasks that a simple user can perform, they can manage the list of users. The latter has been done using the Unified Modeling Language (UML) that allows to schematize the solution as well as the programming languages Angular, Node.js, Ngx-Charts and MySQL as Database Management System (DBMS).

Keywords : *UML, Angular, Node.js, Ngx-Charts, MySQL.*